

Implementation of a power driver controller using rapid control prototyping to control actuators in a medium speed internal combustion engine

Magnus Enqvist

Master's thesis

Supervisors: D. Sc. Jari Hyvönen, Wärtsilä Finland Oy

M. Sc. Tom Kaas, Wärtsilä Finland Oy

B. Sc. Tom Troberg, Wärtsilä Finland Oy

Examinator: Prof. Margareta Björklund-Sänkiahö

Energy Technology, Vasa

Study programme in Chemical Engineering

Faculty of Science and Engineering

Åbo Akademi University

September 24, 2018

ABSTRAKT

Rapid control prototyping är ett nytt koncept som används för att snabbt utveckla och testa prototyper för styrsystem. Rapid control prototyping har varit ett grundläggande element i de tester som har utförts för detta diplomarbete.

Det centrala i detta diplomarbete har varit att undersöka drivsteg för bränsleinsprutning för att bestämma om det finns drivsteg till salu hos en tredje part, som skulle möta de krav som ställdes för detta arbete. Den andra delen av arbetet var att utveckla ett styrsystem som kan användas när det valda drivsteget ska testas.

Matlab Simulink användes för att utveckla styrsystemet. Simulink är ett datorprogram utvecklat av MathWorks och som gör det möjligt att skapa grafiska simuleringsmodeller. För testningen av styrsystemet användes realtidshårdvara utvecklad av Speedgoat, och utsignalerna från systemet mättes med hjälp av ett oscilloskop.

Resultaten av denna undersökning gav en sammanfattning av tillgängliga tredje parts drivsteg och ett fungerande styrsystem som har testats med realtidshårdvaran. Styrsystemets noggrannhet har även testats och analyserats.

I framtida forskning kunde en mera utförlig undersökning av de fel och abnormaliteter som har observerats under testningen av styrsystemet genomföras. Ett test där det utvecklade styrsystemet används för att styra Wärtsiläs nuvarande drivsteg kunde också göras så att kompatibiliteten mellan dem kunde fastställas.

Nyckelord: Rapid control prototyping, Simulink, styrsystem, drivsteg

ABSTRACT

Rapid control prototyping (RCP) is a relatively new concept which is a development and test methodology that can be used to quickly design control system prototypes. RCP has been a key element in the testing phase of this thesis.

The main goals of this thesis were to perform an investigation into the market of injection drivers to establish which alternatives are available to be used in the study, within the limitations and requirements set up for it. And also to develop a control system that is compatible with the injection driver used in the study.

For the development of the control system, Matlab Simulink was used. Matlab Simulink is a program created by MathWorks, which enables the creation of graphical simulation models. A Speedgoat real-time target machine was used as the means for testing the developed control system and the outgoing signals were measured with an oscilloscope.

The thesis resulted in a summary of the availability of injection drivers that fit the requirements established for the driver, and also a fully functioning control system that has been tested with a real-time target machine where the accuracy of the demand and timing of the injection pulses has been measured and analyzed. Future research into the elements of this thesis could focus on a more thorough investigation into the errors and abnormalities that were discovered during the testing of the control system and performing a test where the control system is used to control Wäertsilä's existing injection driver CCM-30 to establish the compatibility between the two.

Keywords: Rapid control prototyping, Simulink, Control system, Injection driver

TABLE OF CONTENTS

Abstrakt	II
Abstract	III
Table of contents	IV
List of symbols and abbreviations	VI
1 Introduction	1
1.1 Background and purpose	1
1.2 Disposition	2
2 Literature review / Theory	3
2.1 Fuel injection	3
2.1.1 Air/Fuel ratio	3
2.2 Solenoid actuator	5
2.2.1 Electromagnetism	6
2.2.2 Ampere's law	6
2.2.3 Magnetic force and electrical time constant of a solenoid actuator	8
2.3 Multiple fuel injections in modern diesel engines	10
2.4 Engine control unit	11
2.5 Electrical control systems	11
2.6 Application specific integrated circuit	11
2.7 Field-Programmable Gate Array	12
2.8 Control systems	14
2.8.1 History	15
2.9 Rapid control prototyping	16
2.9.1 Embedded product development	17
2.9.2 Overview of the rapid control prototyping process	18
2.9.3 Real-Time system	19
3 Method and laboratory test	20
3.1 Building the control system	20
3.1.1 Position measurement and reset pulse	20
3.1.2 Demand, timing and cylinder subsystems	22
3.1.3 Control loop for activating the fuel injection	24
3.2 Importing the Simulink model to Speedgoat	27
3.3 The setup of the testing rig	28
4 Results	31
4.1 Accuracy of the offset of the cylinders	31
4.2 Accuracy of the demand length of the cylinders	35
4.3 Third party injector driver study	38
5 Discussion and assessment of the results	39
5.1 Simulation of the engine control system	39
5.2 Third party injector driver study	40
6 Conclusions and recommendations	41

Swedish summary.....	43
7 References	48

LIST OF SYMBOLS AND ABBREVIATIONS

Scientific

B	Magnetic flux density
CO	Carbon monoxide
CO ₂	Carbon dioxide
°CA	Degrees crank angle
Φ	Magnetic flux
F_{mag}	Magnetic force
F_p	Packing factor
H	Magnetic field intensity
<i>I</i>	Current
J	Current density
L	Inductance
l	Pathlength of vector
<i>l</i>	Length of conducting coil wire
l_T	Length of each turn in conducting coil wire
λ	Air/fuel ratio
λ_a	Relative air supply
λ_f	Relative fuel supply
m_a	Mass of air
$m_{a,th}$	Theoretical mass of air
m_f	Mass of fuel
$m_{f,th}$	Theoretical mass of fuel
μ_0	Permeability of free space in vacuum or air
μ_s	Finite permeability
ms	Milliseconds
μs	Microseconds
<i>N</i>	Number of turns
∇	Del operator

\mathcal{P}	Permeance
p_0	Normalized pressure
R	Resistance
ρ_0	Normalized air density
rpm	Revolutions per minute
S	Surface
S_c	Cross-sectional area
σ	Conductivity
τ_e	Time constant
V_d	Displacement volume

General

ASIC	Application specific integrated circuit
BDC	Bottom dead center
CI	Compression ignition
CCM-30	Cylinder control module 30
ECU	Engine control unit
FPGA	Field-programmable gate array
FPLD	Field-programmable logic device
HILS	Hardware in the loop simulation
IC	Integrated circuit
MBD	Model based design
MILS	Model in loop simulation
PLD	Programmable logic device
RCP	Rapid control prototyping
ROM	Read-only-memory
SI	Spark ignition
SILS	Software in loop simulation
TDC	Top dead center

1 INTRODUCTION

1.1 Background and purpose

With more and more demanding regulations for the reduction of harmful emissions, such as the *EPA 2015* for marine and locomotive engines and *IMO 2016 Tier III* for marine engines. This have led to a need for developing methods to reduce the emission of nitrogen-oxides, unburned hydrocarbons, sulfur-oxides and carbon monoxide. (Zheng, et al., 2014)

One approach to this problem is to render the combustion process in the engines more efficient, thus reducing the emissions. In order to optimize the combustion, a key component is the fuel injection system, since improvement of the fuel injection is a must when trying to approach ideal combustion. One way of achieving this is to improve the electrical system that controls the solenoid actuators, which in turn control the injectors.

In an electrical system the delay between the components is always a factor that needs to be considered, and less delay is always to be preferred. The aim of this thesis is to try and reduce the delay that is present in the communication chain from the control software to the injector driver. If it is possible to reduce the delay to be essentially non-existent, this will help immensely by eliminating any errors that might occur due to the signals becoming un-synchronized as a result of delay.

The need for this study arose when the Advanced Concepts team at Wärtsilä wanted a flexible and easily configurable control system that would handle very fast injections, which the injectors themselves necessarily are not adapted to. Therefore no readily available solution was accessible for this request.

The important parameters that need to be considered for an injector driver in this type of application are output voltage, both peak and hold levels of the driving current and the time for which the peak and hold current can stay activated without damage to the

system. The general purpose of the thesis is to investigate the possibilities for a fast, direct-control application solution for controlling an injector driver. The research questions and goals for the study can be summed up into the following points:

- Are there any third-party injection drivers available on the market that would suit the needs of this project?
- If no third-party injection drivers are available, is it possible to use Wärtsilä's existing drivers for the project?
- Develop a control system for the injection driver in Simulink and test the control system on the Speedgoat real-time target machine.
- Is it feasible for Wärtsilä to start using the new system on a larger scale?

1.2 Disposition

Chapter 2 outlines the theory on the hardware and software relevant for this thesis work. In chapter 3 the methodology and tools used in the project are presented. In chapter 4 the results are presented, and the results are then discussed and evaluated in chapter 5. In chapter 6 the final conclusions and recommendations for future research are presented.

2 LITERATURE REVIEW / THEORY

2.1 Fuel injection

The fuel injection systems task as described by (Heywood, 1988) is to measure the appropriate amount of fuel and then inject it into all the cylinders for each cycle. The amount of fuel injected is determined by the speed and load of the engine. It is of utmost importance that the beginning and end of the injection is precise and clean as to avoid any leakage or secondary injections.

A major improvement in injection technology was the addition of solenoid actuated common rail injection systems which are progressively replacing the stiffly calibrated camshaft systems (Posey, et al., 2012). These systems offer individual control of a number of parameters such as injection timing, pressure and the amount of pulses per cycle as well as the amount of fuel injected with each pulse. These improvements that (Satkoski, et al., 2011) have described have led to a considerable reduction in particulate matter and nitrogen-oxides emissions.

2.1.1 Air/Fuel ratio

The air/fuel ratio is essential for the combustion process in internal combustion engines. There are several methods and phenomenon that affect the amount of air (m_a) that flows to the cylinders. Throttling the supply of air and aerodynamic resistance are an example of such methods described by (Kiencke & Nielsen, 2010). The relative air supply is given by the ratio of real to theoretical value:

$$\lambda_a = \frac{m_a}{m_{a,th}} \quad (2.1)$$

The same conditions apply for the relative fuel supply:

$$\lambda_f = \frac{m_f}{m_{f,th}} \quad (2.2)$$

The amount of fuel which is needed to achieve an ideal stoichiometric combustion of gasoline with a mass of air m_a , is equivalent to the theoretical fuel mass $m_{f,th}$.

$$L_{st} = \frac{m_{a,th}}{m_{f,th}} = 14.66 \quad (2.3)$$

Where $m_{f,th}$ is the theoretical fuel mass and $m_{a,th}$ is the theoretical air mass. The air/fuel ratio lambda is given by:

$$\lambda = \frac{\lambda_a}{\lambda_f} \quad (2.4)$$

Equation (2.4) can be extended and then gives:

$$\lambda = \frac{m_a}{m_f} \cdot \frac{m_{f,th}}{m_{a,th}} = \frac{1}{L_{st}} \cdot \frac{m_a}{m_f} \quad (2.5)$$

If the combustion is stoichiometric the value of lambda is one ($\lambda = 1$) and if lambda is greater than one ($\lambda > 1$) more air is used than needed, this gives a “lean” fuel mixture. Reversely if lambda is less than one ($\lambda < 1$) more fuel is used than needed which gives a “rich” fuel mixture. (Kiencke & Nielsen, 2010). The stoichiometric mixture is fuel composition dependent. A visualization of the differences between stoichiometric, lean and rich fuel mixtures for gasoline is shown in Figure 2.1.

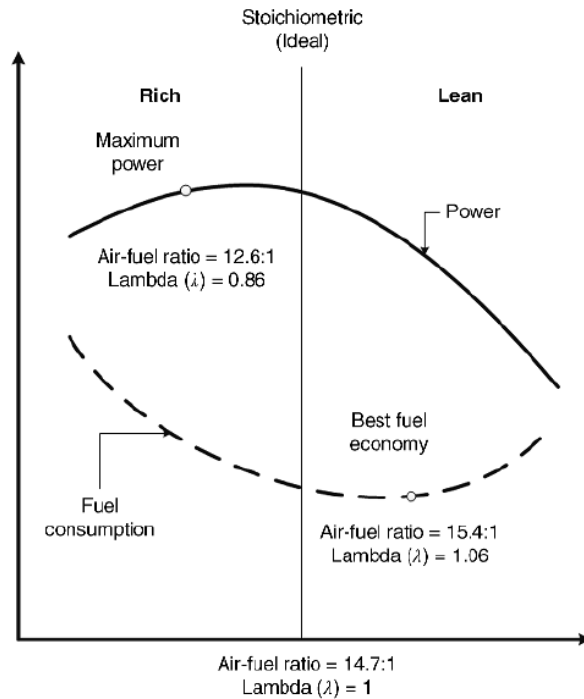


Figure 2.1 a representation of the ideal lambda value and a description of what happens if one goes under or over that value (*Khajepour, et al., 2014*)

2.2 Solenoid actuator

The functionality principle of a solenoid actuator is that it uses magnetic fields in order to transform electrical energy into mechanical energy, thus a solenoid actuator is a form of magnetic actuator.

An essential part of solenoid actuators is the solenoidal coil, which is a coil of wire that is wound into a cylindrical shape inside an iron core. The core has two main parts, which are the armature and the stator. The armature is the moving part while stator like the name hints at is the static part. When the solenoid actuator is activated a magnetic force draws the armature towards the stator which can be likened to a clapping hand, which is why this type of armature is called a clapper armature and an example of such an actuator can be seen in Figure 2.2.

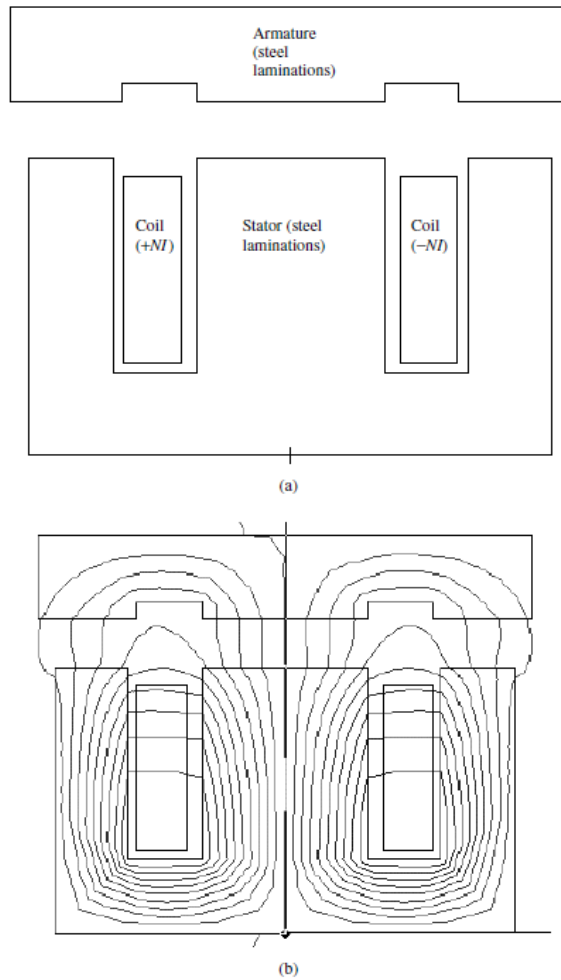


Figure 2.2 a typical solenoid actuator with a clapper armature with the flux lines (b) which are obtained by finite-element analysis. (Brauer, 2006)

2.2.1 Electromagnetism

The field of electromagnetism incorporate both electricity and magnetism. In the electromagnetic interactions there are particles involved which have an electric charge, this attribute is as essential as mass. When particles with mass are accelerated by gravitational forces the particles which are electrically charged are accelerated by electric forces. (Young, et al., 2011)

2.2.2 Ampere's law

The origin of magnetic fields can be described by Ampere's law, which states that for any given point in space the curl of the magnetic field intensity \mathbf{H} equals the current density \mathbf{J} . (Brauer, 2006)

$$\nabla \times \mathbf{H} = \mathbf{J} \quad (2.6)$$

Where \mathbf{H} is magnetic field intensity in amperes per meter, and \mathbf{J} is the current density in amperes per square meter. (Brauer, 2006) In air, the magnetic flux density \mathbf{B} is related to \mathbf{H} by the equation.

$$\mathbf{B} = \mu_0 \mathbf{H} \quad (2.7)$$

Where \mathbf{B} is the magnetic flux density in teslas and μ_0 is the permeability of free space in vacuum or air. The field intensity \mathbf{H} “rotates” around a current axis in the same way that for example a wheel rotates around its axis, in the case of a solenoid the current axis is the coil. The ampere’s law equation (2.6) can be expressed in another way by integrating it over a surface \mathbf{S} . (Brauer, 2006)

$$\int (\nabla \times \mathbf{H}) \cdot d\mathbf{S} = \int \mathbf{J} \cdot d\mathbf{S} \quad (2.8)$$

Where \mathbf{S} is a surface in square meters. Stoke’ law can be used to replace the surface integral by a line or closed path integral, which then gives the expression which is the most commonly used integral form of Ampere’s law. (Brauer, 2006)

$$\oint \mathbf{H} \cdot d\mathbf{l} = \int \mathbf{J} \cdot d\mathbf{S} = NI \quad (2.9)$$

Where \mathbf{l} is the pathlength of the vector in meters, and the circle on the integral sign indicates that the path is closed. The total current can also be written as the product of the number of conductors (turns) N times the current I . (Brauer, 2006)

If magnetic flux density \mathbf{B} is integrated over any surface \mathbf{S} the result will be magnetic flux. Since the unit of \mathbf{B} are teslas which is the same as webers per square meter the unit for magnetic flux is webers. (Brauer, 2006)

$$\Phi = \int \mathbf{B} \cdot d\mathbf{S} \quad (2.10)$$

An essential characteristic of magnetic flux is what happens if \mathbf{S} is a closed surface, then the total amount of magnetic flux that flow through the closed surface is zero. A closed surface is any surface that entirely encloses a volume. The formula for the flux through a closed surface then becomes the following. (Brauer, 2006)

$$\Phi = \oint \mathbf{B} \cdot d\mathbf{S} = 0 \quad (2.11)$$

Divergence is defined as net output flux per unit volume, when the zero flux present in equation (2.11) is applied to a very small volume with a closed surface it gives the following equation. (Brauer, 2006)

$$\nabla \cdot \mathbf{B} = 0 \quad (2.12)$$

Consequently the magnetic flux density is always divergenceless. Since $\mathbf{B} = \mu_0 \mathbf{H}$ as can be seen in equation (2.6) it can be concluded that in air \mathbf{B} just as \mathbf{H} will “rotate” around the current axis.

2.2.3 Magnetic force and electrical time constant of a solenoid actuator

The most important parameter for a solenoid actuator is the magnetic force, which determines the response speed and mechanical acceleration of the actuator. The equation for the magnetic force on steel is the following. (Brauer, 2006)

$$F_{mag} = \frac{SB^2}{2\mu_0} \quad (2.13)$$

In equation (2.12) the steel is assumed to have infinitely high permeability. In the case of the steel having finite permeability μ_s equation (2.12) will get an additional force term. (Brauer, 2006)

$$F_{mag} = S \left(\frac{B^2}{2\mu_0} - \frac{B^2}{2\mu_s} \right) \quad (2.14)$$

Another important parameter for the solenoid actuator is the time constant. The currents rising time rate is controlled by the inductance L and the resistance R. Thus the time constant τ_e for the system is given by the equation. (Brauer, 2006)

$$\tau_e = \frac{L}{R} \quad (2.15)$$

The inductance L is proportional to the product of the permeance \mathcal{P} and the square of the number of turns in the coil N. (Brauer, 2006)

$$L = N^2 \mathcal{P} \quad (2.16)$$

The resistance in the coil wire R in ohms is related to the length of the wire. In the following equation the length of the conducting coil wire l is divided by the conductivity σ and the cross-sectional area of the wire S_c . (Brauer, 2006)

$$R = \frac{l}{\sigma S_c} \quad (2.17)$$

If the length of the wire is expressed as the length of each turn in the wire l_T times the number of turns in the coil N, the equation for the coil resistance R can be rewritten in the following way. Where the packing factor is expressed as F_p . (Brauer, 2006)

$$R = \frac{N^2 l_T}{\sigma F_p S_c} \quad (2.18)$$

Then by using equation (2.16) and (2.17) it is possible to rewrite the time constant equation (2.15) to get the time constant of a solenoid actuator. (Brauer, 2006)

$$\tau_e = \frac{\mathcal{P}l_t}{\sigma F_p S_c} \quad (2.19)$$

As can be seen the time constant for the solenoid actuator is not dependent of the number of turns N in the coil. (Brauer, 2006)

2.3 Multiple fuel injections in modern diesel engines

According to (Lucchini, et al., 2010) the combustion efficiency as well as the in-cylinder emission control in diesel engines are two parameters which heavily depend on the injection strategies, specifically the effectiveness of them. This has led to the development of new advanced technologies when it comes to fuel injection equipment, which will allow for more precise and flexible control of many parameters such as, injection timing, fuel delivery, injection rate, injection pressure and multiple injections. (Chen, 2000)

The definition of a multiple injection is according to (Chen, 2000), an injection strategy where the injection of the fuel takes place prior to the combustion stroke or during it and the injection is spaced out over at least two separate injections. There are a number of different multiple injection strategies which are commonly used, pilot injection and post injection are two of them. In pilot injection the injection of a small amount of fuel is done before the main injection, and in post injection a small amount of fuel is injected right after the main injection. (Chen, 2000)

The main benefits of multiple injections and more specifically pilot injection are according to (Okude, et al., 2007) reduced noise in the combustion and a reduction of Nitrogen-oxide (NOx) emissions. The reason for this is that the pilot injection shortens the main injections ignition delay period and also limit the quantity of the initial premixed combustion.

2.4 Engine control unit

An engine control unit (ECU) is a breed of electronic control unit that is used to control actuators in an internal combustion engine, the ECU is essential for better engine efficiency (Mendonca, et al., 2017).

The structure of the ECU as it is known today got its beginning during the 80's and the overall concept of it looks the same today. However, some advancements have been made with the introduction of new technologies such as application specific integrated circuits (ASIC) and field-programmable gate arrays (FPGA) which have notably improved the effectiveness of the ECU. (Farrugia, et al., 2015)

Some of the main parameters that the ECU have control over and monitor are, air-fuel ratio, variable valve timing, spark or fuel injection timing and exhaust gas recirculation. In order to do this the ECU is connected to a large number of sensor that measure for instance the cylinder pressure, air flow, temperature, crankshaft position and cylinder knock. (Breitzman, 1985)

The control system that is developed in this study can be described as a sort of scaled down version of an ECU that only measures the speed and position of the engine and controls the injection timing pulses.

2.5 Electrical control systems

In the following chapters some of the electrical components and control systems that are essential for this study are examined and presented, such as application specific integrated circuits and rapid control prototyping systems.

2.6 Application specific integrated circuit

The definition of an application specific integrated circuit (ASIC) is an integrated circuit (IC) that is designed for a specific application or purpose (Hilbert, 1991). This diverges quite a bit from the standard IC products that are designed to be able to be

used in a wide range of different applications. Examples of these standard IC products are microprocessors and memories of different sorts.

The ASICs history began with the masked read-only-memory (ROM), but it wasn't until the 1980s that the use of ASICs started to be the dominant technology in the IC marketplace around the world. (Hilbert, 1991)

There are essentially four broad categories into which ASICs can be divided, these are:

- Full-custom
- Semi-custom
- Field-programmable logic devices (FPLDs)
- Linear arrays

Field-programmable gate arrays are found in the FPLDs category in the ASIC taxonomy, and it was the increasing popularity of the FPGAs that moved the FPLDs from being standard products into being classified as ASICs. (Hilbert, 1991)

2.7 Field-Programmable Gate Array

A Field-Programmable Gate Array (FPGA) is a multi-level Programmable Logic Device (PLD) with a very high logic capacity thanks to its general structure (Brown & Rose, 2002). The structure of FPGAs are built up of blocks of logic which are connected to each other by programmable interconnects, a typical FPGA structure is illustrated in Figure 2.4. It is the programmable interconnects which enables the implementation of multi-level logic by the users, this in turn gets rid of a number of size limitations which derives from the two-level logic structure of PLDs. (Trimberger, 1994)

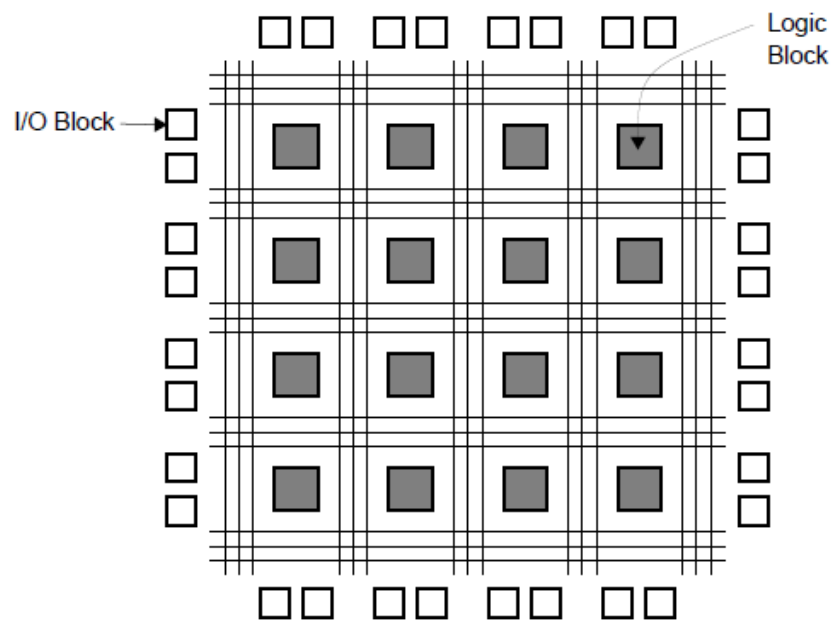


Figure 2.4 Illustration of a typical FPGA structure. (Brown & Rose, 2002)

The number of blocks and their size and structure as well as the extent and connectivity of the programmable interconnects differ significantly within the different FPGA architectures. The driving force behind these differences in architecture is the different programming technologies and also which target applications the parts are made for. (Trimberger, 1994)

The main difference between a FPGA and a CPU is that the hardware configurability is of a considerably higher degree in the FPGA as described by (Underwood, 2004). Because of this the designers of the CPU must choose a memory hierarchy and resource allocation that will perform properly with many different applications. The designers of the FPGA can on the other hand leave many of these decisions to the designers of the applications.

In terms of peak floating-point performance the FPGA outperforms the CPU, this is because the FPGA can customize the resource allocation for the specific needs of the application, this while the CPU has fixed functional units. On the other hand the CPU has a higher peak performance compared to the FPGA in applications that makes full use of the fixed resources. (Underwood, 2004)

2.8 Control systems

A control system in its most simple form can be described by the following three components:

- Actuating signals or inputs
- Control-system components
- Controlled variables or outputs

The link between these points and what can be described as the working principle of the control system is that the system controls the outputs in a way that is described by the inputs and this is done through elements of the control system.

There are essentially two different broad categories of control systems, open-loop systems and closed-loop systems. The systems each have their own strengths and weaknesses, so they are usually suitable in different scenarios.

The open-loop systems main advantage is that it is simple and low-cost, the disadvantage however is that it usually is an inaccurate system. The reason for this is that the open-loop system lacks a way of knowing if the output is changing and therefore can't adjust the signal according to the change in the output. This leads to open-loop systems mainly being used in noncritical applications. In Figure 2.5 a block schematic for a standard open-loop system is depicted. (Golnaraghi & Kuo, 2009) (Bishop & Dorf, 2011)

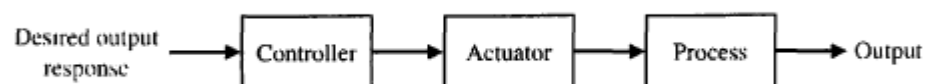


Figure 2.5 Representation of an open-loop system using block schematics (Bishop & Dorf, 2011)

The difference between a closed-loop system and an open-loop system is that the closed-loop system is able to measure the output signal and then through one or multiple feedback paths send that signal back to be compared with the input. This

makes the closed-loop system more accurate than the open-loop system, since it can with the help of the feedback signal correct the error between the reference input and the actual measured output. A block schematic for a typical closed-loop system is presented in Figure 2.6. (Nise, 2011)

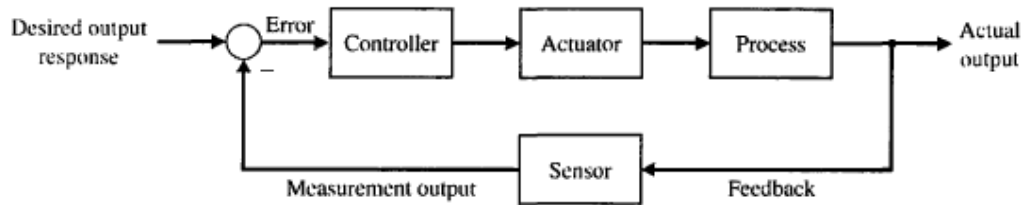


Figure 2.6 Representation of a closed-loop system using block schematics (Bishop & Dorf, 2011)

2.8.1 History

The use of automatic control and especially the use of feedback control has a very long history with its first appearance being in Greece during 300 to 1 B.C with the advancement of a float regulator mechanism. This float regulator design was used for maintaining a constant level of liquid in various process. In the first century A.D. in Alexandria a book by the name of *Pneumatica* was published by Heron, this book contained designs of several water-level mechanisms which used float regulators.

In modern times in Europe the first control system with feedback that was developed was a temperature regulator invented by Cornelis Drebbel (1572-1633). James Watt developed in 1769 the flyball governor which is considered to be the first automatic controller with feedback that was used in an industrial process. The flyball governor which was an all-mechanical device was used to control the speed of a steam engine. This was done by measuring the output shaft's speed and then using the momentum of the flyball to regulate the flow of steam entering the engine, the design of the flyball governor is depicted in Figure 2.7. (Bishop & Dorf, 2011)

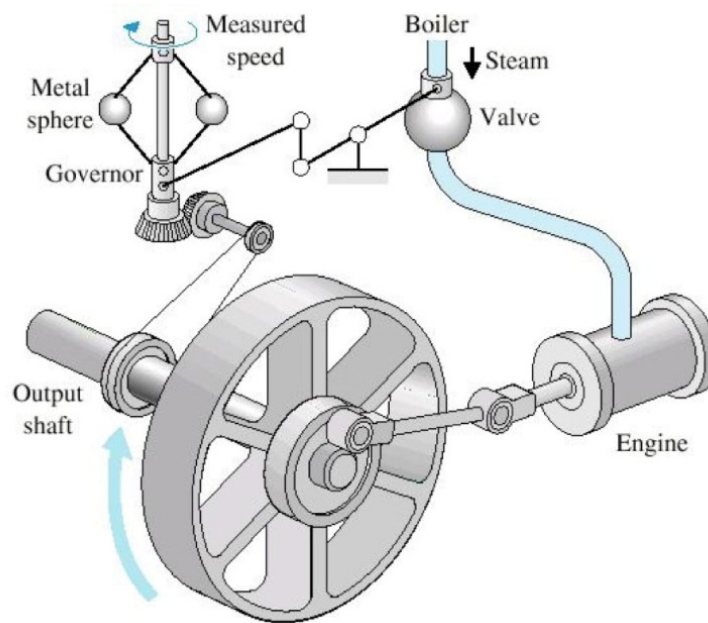


Figure 2.7 Sketch of James Watt's flyball governor (*Bishop & Dorf, 2011*)

The basic working principle of the flyball regulator can be loosely associated with the mechanical fuel injection systems used in internal combustion engines. This then means that for over 200 years, roughly the same principle has been used until the electrical injection systems started to take over.

2.9 Rapid control prototyping

Rapid Control Prototyping (RCP) can be described as a development and test methodology used for quickly designing control system prototypes to be tested with the actual physical hardware (Young, et al., 2017). This simplifies the design process since it is possible to early on verify the control algorithms before beginning the embedded implementation stage.

2.9.1 Embedded product development

RCP is the third step in the embedded product development chain. There are five steps altogether which are illustrated in Figure 2.8. (Khan, 2016)

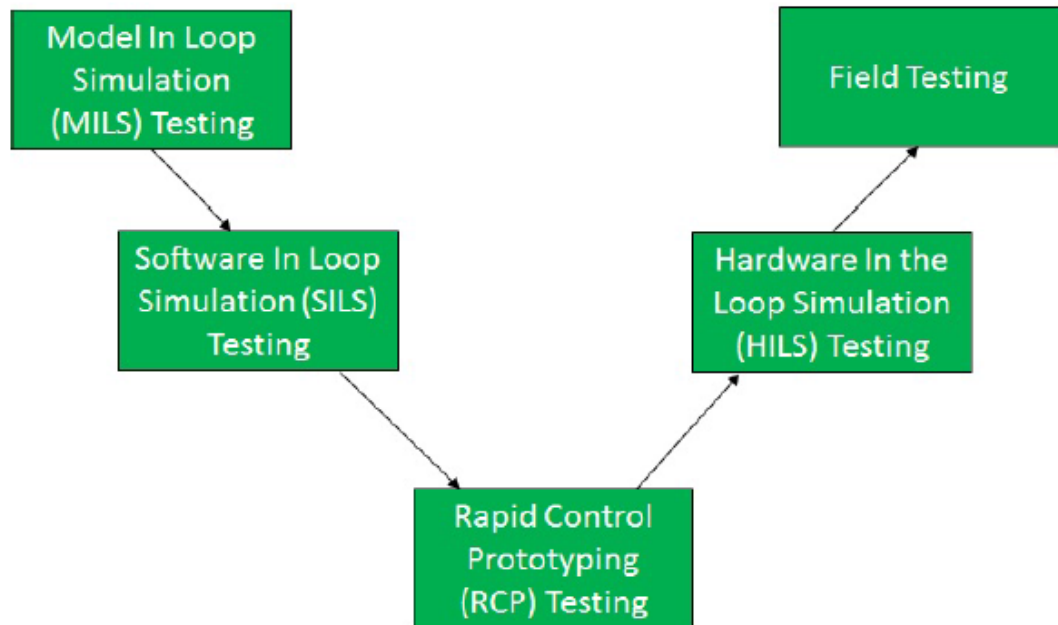


Figure 2.8 Validation stages of embedded product development (Khan, 2016)

In the first step of embedded product development the core algorithm is validated by using Model in Loop Simulation (MILS). MILS is a process where Matlab Simulink, or similar software tools, is used to simulate actuators and sensors in a closed loop and the core algorithm in a different Simulink model.

The next step is Software in the Loop (SILS) simulation where Matlab Simulink is using auto code generation to generate C code of the models of the actuators and sensors and then validating them. According to the work of (Khan, 2016) these first two steps can be categorised as Model Based Design (MBD) which is a technology that is widely used in embedded product development.

Rapid Control Prototyping is the third step in the embedded product development process, it consists of testing the algorithm on a real time platform in a closed loop

simulation and using real actuators and sensors. After RCP the next step is Hardware in the Loop Simulation (HILS) which is where the application is available in a more stable and mature form in an embedded hardware system. The actuators and sensors are then simulated in this system with a set of hardware and software that is synchronized, which then validates the accuracy and the behaviour of the algorithm. The final step in the validation chain is field testing where the finished software is implemented and tested in the actual process or area in which it will be implemented in when it is finished. (Khan, 2016)

2.9.2 Overview of the rapid control prototyping process

In short, the process of rapid control prototyping consists of the five following phases:

- Making a model in a simulation environment of the process that will be controlled
- Validating the model
- Making a prototype of the control system
- Combining the prototype of the control system with the model of the process and testing them against each other
- Testing the prototype with the real hardware of the process to be controlled

In their research (Hölttä, et al., 2004) identifies two different basic approaches of RCP, these are full pass and bypass. In the full pass approach, the current control system is replaced altogether by the new designed one, which then has full control over the original process. With the bypass method the current control system and the new one works side by side, with the new system having control over for example any new actuators or processes to be implemented and the current system having control over the ones that already exist.

2.9.3 Real-Time system

A loose definition of a real-time system is a system in which the response time is an important factor for the system to function correctly, if the response time does not satisfy the bounded time constraints there is a risk of failure. The logical correctness of a real-time system is established by the timing and correctness of the outputs. (Krishna, 1999) (Laplante & Ovaska, 2012)

There are also two general classifications of real-time systems which are:

- Hard Real-Time systems – system where the time deadlines need to be fulfilled or there is a high chance of damage. Examples of such systems are: aircraft autopilot and ABS on a car.
- Soft Real-time systems – if the time deadlines of the system is not fulfilled there is only a decrease in the systems performance. Examples of such systems are: video conference equipment and the operating system on a computer. (Greppl, 2011)

In which of these categories a real-time system belongs is often based on a subjective opinion of the person making the decision since many real-time systems can be placed in either one according to the right supporting scenario. A common misconception of real-time systems is that they are synonymous with a fast system which is not the case, although many hard real-time systems have time deadlines which can be as small as tens of milliseconds. (Laplante & Ovaska, 2012) (Greppl, 2011)

With these definitions in mind the system which will be developed in this study will most likely be classified as a hard real-time system, but arguments can always be made for the opposite. Arguments for it being a hard real-time system is that critical engine damage can occur if the control system malfunctions during any crucial stage of the cycle. The case for it being a soft real-time system can be made since the risk of personal injuries in direct correlation to the failure of the system is relatively small.

3 METHOD AND LABORATORY TEST

3.1 Building the control system

The control system that was used in this thesis was developed in Matlab Simulink. Matlab is a programming language that was developed by MathWorks. Simulink is a tool that allows the user to build a graphical simulation model by using blocks that contain predefined Matlab code.

For testing of the control system a real-time target machine from Speedgoat was used. Speedgoat is a company that have specialized in the development of real-time systems designed to specifically work with Simulink and the real-time operating system Simulink Real-Time made by MathWorks.

The operating principle for the control system is that it will produce a signal that will control the injector driver in the engine. The control signal will essentially be a digital high–low signal (0-5V). This signal will then determine when the injector driver will activate and output the required voltage and current to control the solenoid actuator in the injector, thus triggering the injection of fuel.

3.1.1 Position measurement and reset pulse

The position measurement in the model is built up in the way that the cycle starts at zero degrees, which is the combustion Top Dead Center (TDC), and then it counts upwards toward 360 degrees which is then the gas exchange TDC. From there it counts down from -360 degrees to 0 degrees again and the cycle is completed. This happens because of the fact that the engine that the system was built for is a four-stroke engine, and thus each combustion cycle has two revolutions. On account of this, the Bottom Dead Center (BDC) will happen at 180 degrees and -180 degrees in the position measurement.

In Figure 3.1 the position/reset subsystem is displayed. The subsystem has one input signal and two output signals. The input signal to the system is the position measurement of the cycle in Deci degrees, and the first output supplies the Stateflow chart with the position of the engine. The input is also divided into two signals, which will be compared to see if they are bigger than zero or smaller than zero respectively. Then these signals are led into an AND logical gate, and the result is that the signal from the AND gate will display a logical one when the position measurement is zero, in other words when the piston is in TDC and the cycle restarts. This is then the reset signal for the stateflow chart.

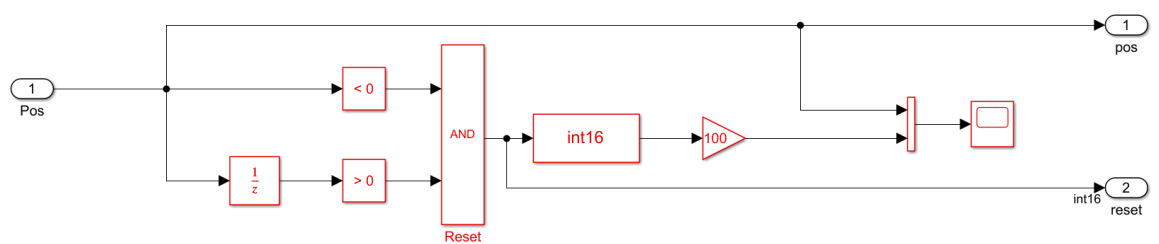


Figure 3.1 Position/reset subsystem in the Simulink model

Each of the cylinders have their own offset in degrees crank angle (°CA) from the position of the engine cycle and these individual offsets can be seen in Table 1 below.

Table 1. Offset of the cylinders

Cylinder	Offset in °CA
Cylinder 1	0
Cylinder 2	480
Cylinder 3	240
Cylinder 4	600
Cylinder 5	120
Cylinder 6	360

3.1.2 Demand, timing and cylinder subsystems

The only parameters that are user defined in the control system are demand and timing. Timing is the degrees crank angle when the fuel injection pulse shall begin, demand is the amount of time that the fuel injection pulse shall last. These two parameters are delivered to the control system in the form of vectors, each vector represents either a row of demand inputs or timing inputs. The values for these parameters are stored in a Matlab m-file from which they are imported when the program is run. The vectors and their outputs are presented in Figure 3.2.

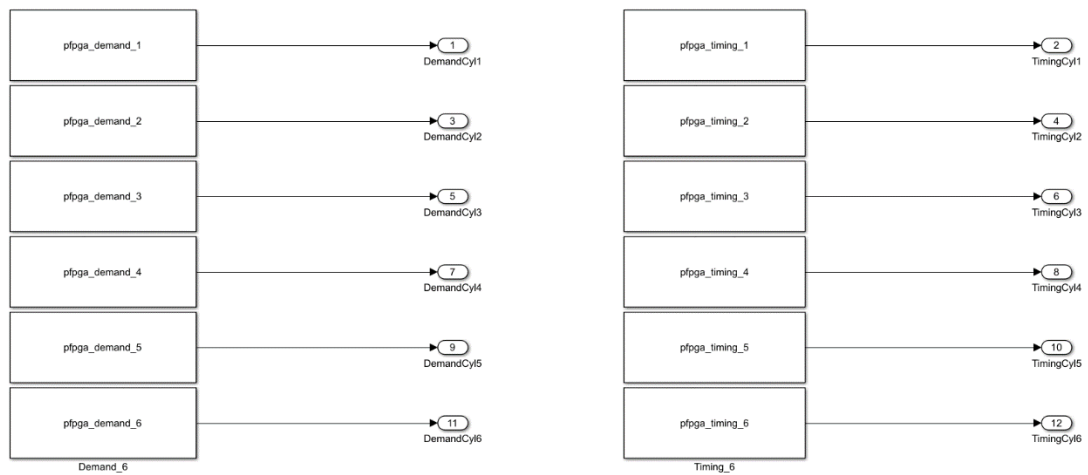


Figure 3.2 Demand and timing vectors used in the model

The next step is that the outputs of the vectors are led into the next system in the model which is the main subsystem that will be transferred to the Speedgoat FPGA card later on in the process. The demand and timing as well as the position of the engine and the position synch are inputs to this subsystem that can be seen in Figure 3.3.

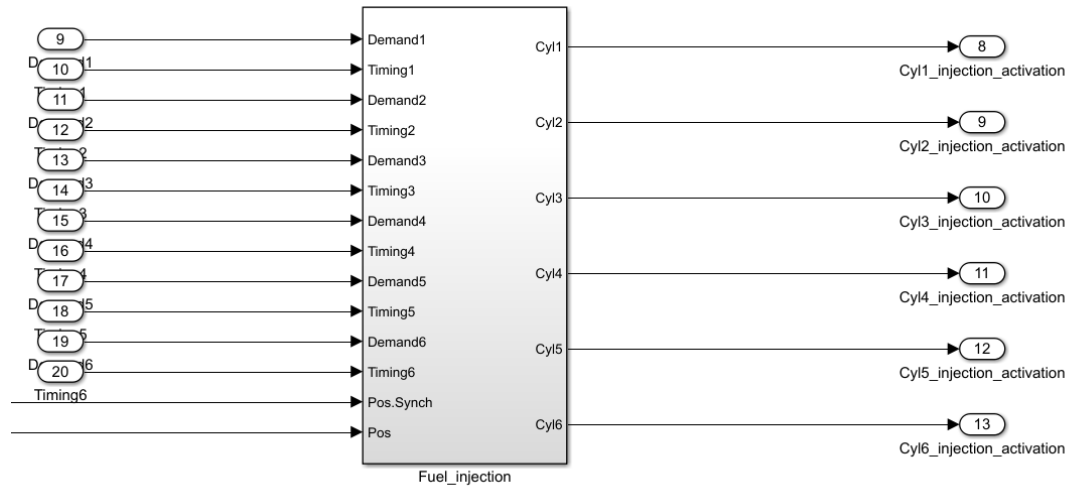


Figure 3.3 Overview of the inputs and outputs of the main control subsystem

Inside of the Fuel_injection subsystem the demand and timing vectors are connected as inputs to their respective cylinder subsystem along with the correct position for that cylinder and the position synch which is the same for all cylinders. In Figure 3.4 these inputs can be seen as well as the outputs which carry the activation signal for the fuel injection.

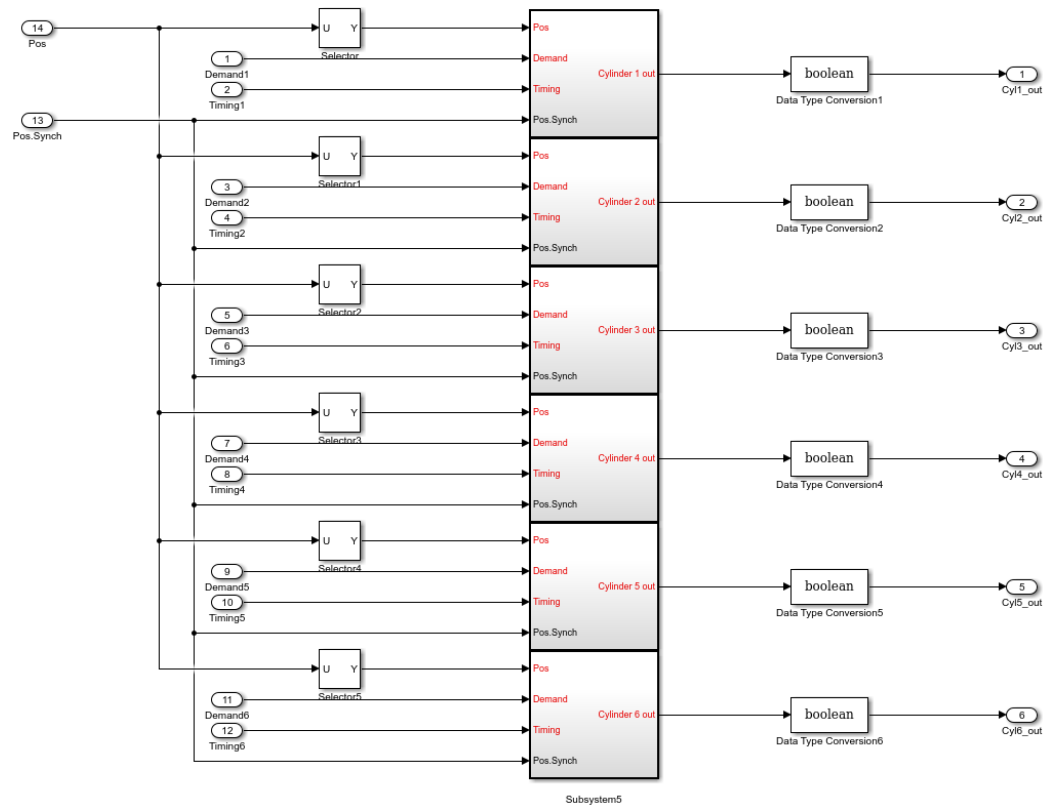


Figure 3.4 Inputs and outputs of the individual cylinder subsystems

Then inside the different cylinder subsystems the demand and timing vectors are divided up into individual signals. This is achieved by having five selector blocks and each of them selects an element from the original signal, thus creating five individual signals. The selector blocks are configured such that the first one will select the first value in the vector and the second one the second value and so on. This is done both for the demand vector and the timing vector through two identical subsystems to the one depicted in Figure 3.5. These ten output signals are then used as inputs into the stateflow chart.

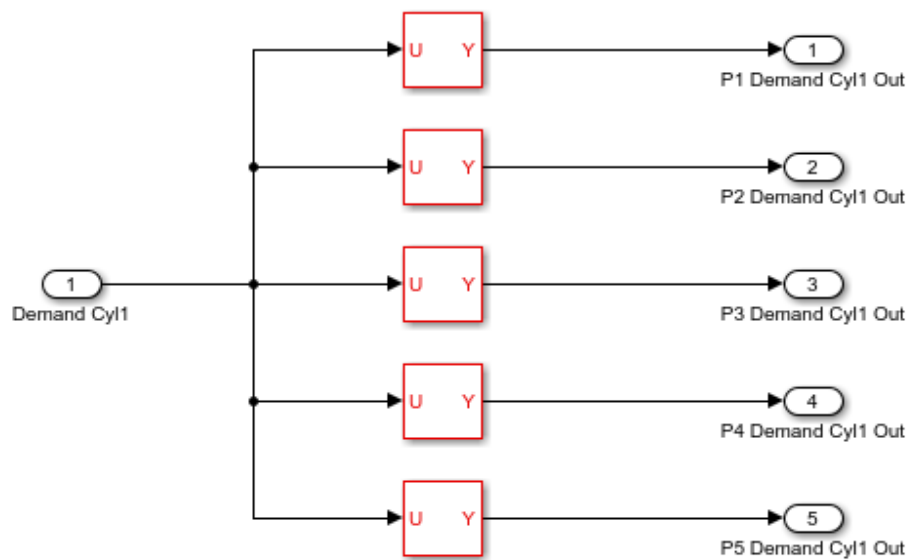


Figure 3.5 Subsystem which splits up the vector into five individual signals.

3.1.3 Control loop for activating the fuel injection

The main control module of the system is the control loop that controls the activation of the fuel injection which is made as a stateflow chart in Simulink. When the simulation starts the stateflow chart begins to run as well with the first state that is depicted in Figure 3.6, which waits for the position synch signal to be activated which means that the position in the program is synchronised with the actual position of the engine.

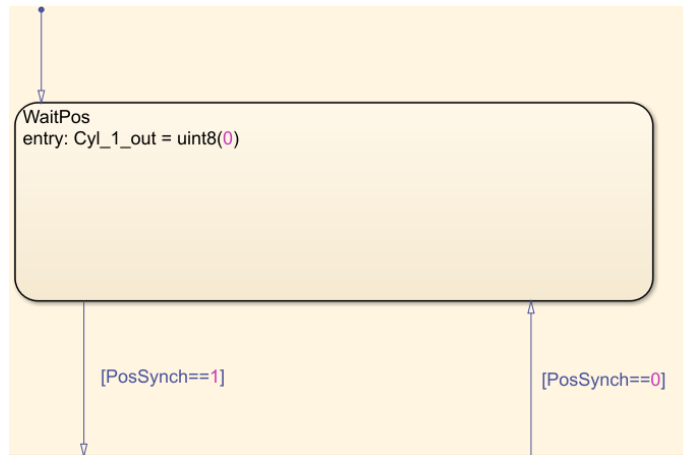


Figure 3.6 Initial step in the stateflow chart program

Then when the position synch is active the program moves on to the next state which is the main control loop inside which the rest of the states are incorporated. The beginning state in the main control loop is then another wait state similar to the previous one. Here the program then waits for the reset signal to be activated, then when that happens the program moves onto the next state which waits for the timing of the first position of the cylinder, which is defined in the timing vector for the cylinder these two states are shown in Figure 3.7. Then when the value of the position of the engine is equal to or greater than the value defined in the timing vector the program moves on to the next state.

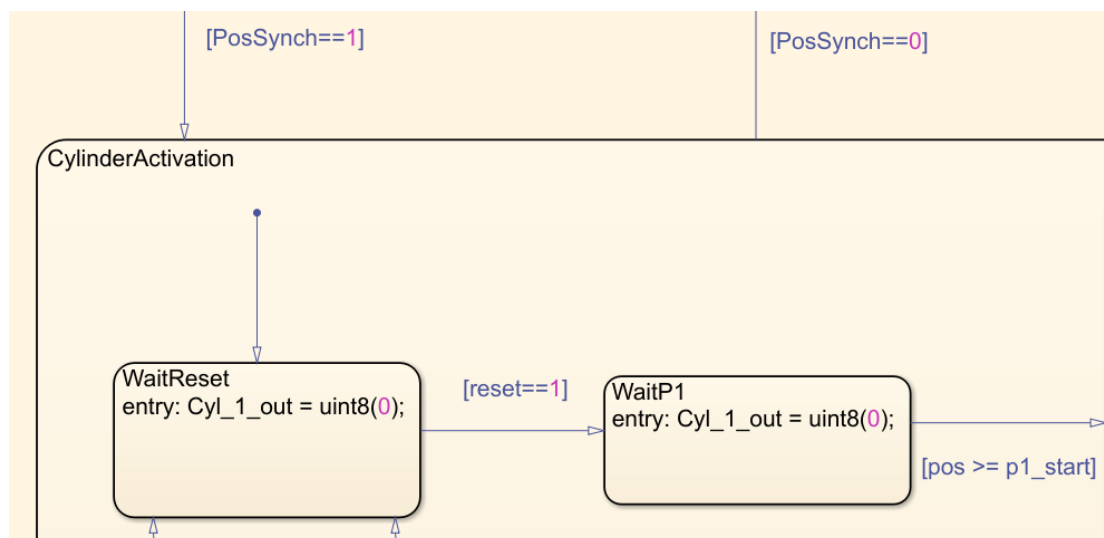


Figure 3.7 Reset state and the first wait state for the timing

The next state depicted in Figure 3.8 then checks if the demand for that timing position is zero or if it has a value, if the demand is zero the program moves immediately onto the wait state for the next timing point of the cylinder. If the demand however is not equal to zero then the next state is activated which puts the output signal to one and then starts counting from zero until it reaches the same value as the one that is defined in the demand vector for the cylinder. Then when that value has been reached the output signal will be set back to zero and the program will move on to the wait state for the next timing point for the cylinder.

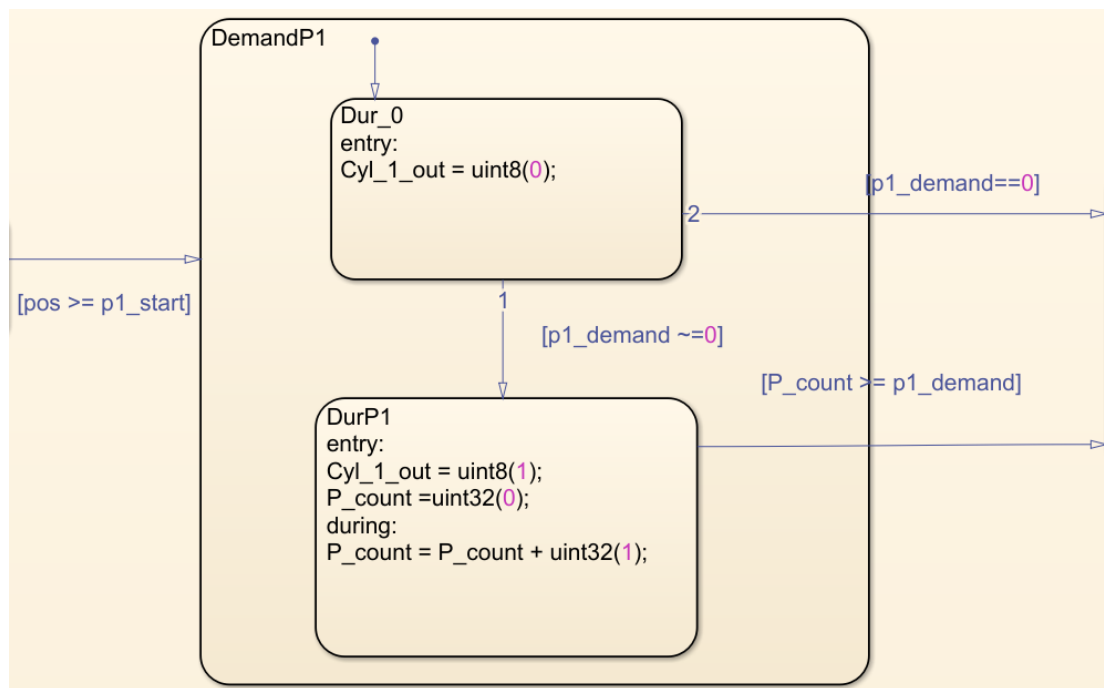


Figure 3.8 State that handles the demand of the injection pulse

This chain is then repeated for all the five injection points for the cylinder and then when it have reached the last demand state of the cycle it moves back to the beginning of the loop and waits for the reset signal to start the cycle again. The only exception is if at any point in the cycle the position synch signal would become zero, then the program would go back to the initial wait state and wait there until the position synch signal would become one again. The state flow chart for the control system in its entirety is depicted in Figure 3.9.

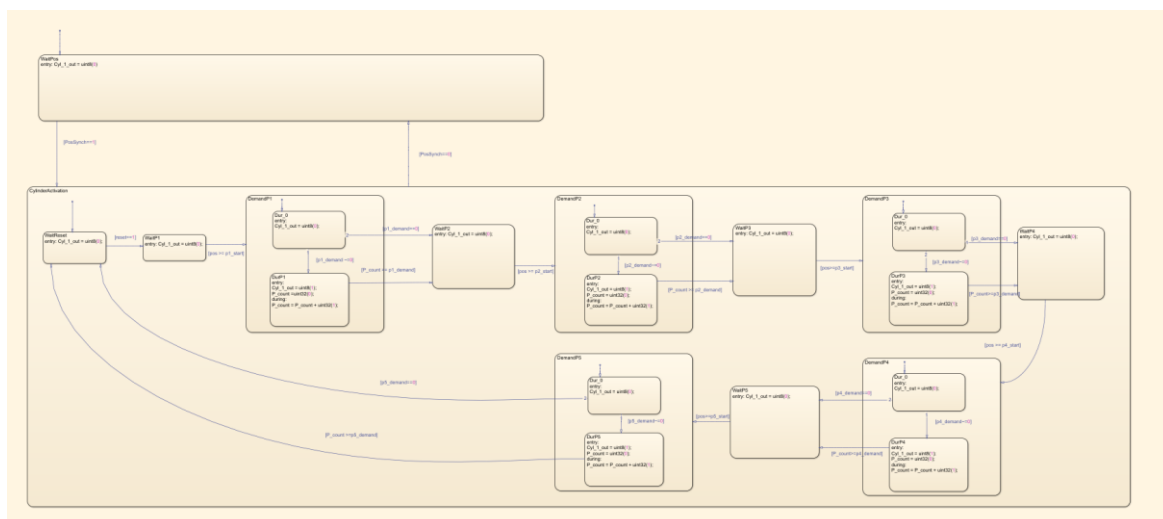


Figure 3.9 State flow chart of the cylinder control system

3.2 Importing the Simulink model to Speedgoat

In order for the fuel injection activation model to be able to be implemented on the Speedgoat FPGA-card a synthesis of the model must be created with the help of the HDL advisor tool in Simulink. The HDL advisor tool converts the model into a format that the FPGA-card in the Speedgoat target machine understands.

Using Matlab a build of the entire Simulink-model, which consists of more parts than the fuel injection activation subsystem is made which will be used as the software in the Speedgoat target machine. The software is then loaded into the target machine via an Ethernet cable by using the custom made tool Wäertsilä Speedgoat Application Loader (WSAL). Then CANape is used to control the parameters when the software runs on the target machine, a picture of the interface in CANape is shown in Figure 3.10.

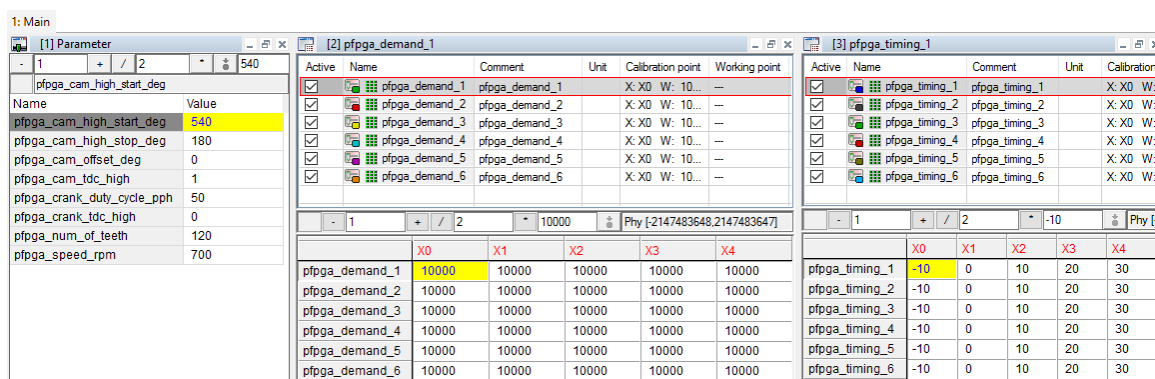


Figure 3.10 Parameter configuration in CANape

3.3 The setup of the testing rig

The Speedgoat real-time target machine is mounted in a cabinet in order to allow for easier accessibility and maneuverability when setting up the tests in an office environment. The part of the target machine that allows it to communicate with external devices can be seen in Figure 3.11. The Ethernet host link is used to connect the target machine to a computer which will run software that will control the target machine, in this thesis CANape was used for this purpose. The I/O333-6 adapter is connected to an external I/O card on the other side of the cabinet and that card is used to get inputs and outputs from the program that is implemented on the target machine.



Figure 3.11 Picture of the target machine connection setup

The I/O333-6 card that was used in the testing can be seen in Figure 3.12, where all the channels that were used are marked out with their respective functions. As can be seen the cylinders used in the moment the picture was taken is cylinder five and six as well as the speed and phase of the simulated engine which always was in use.

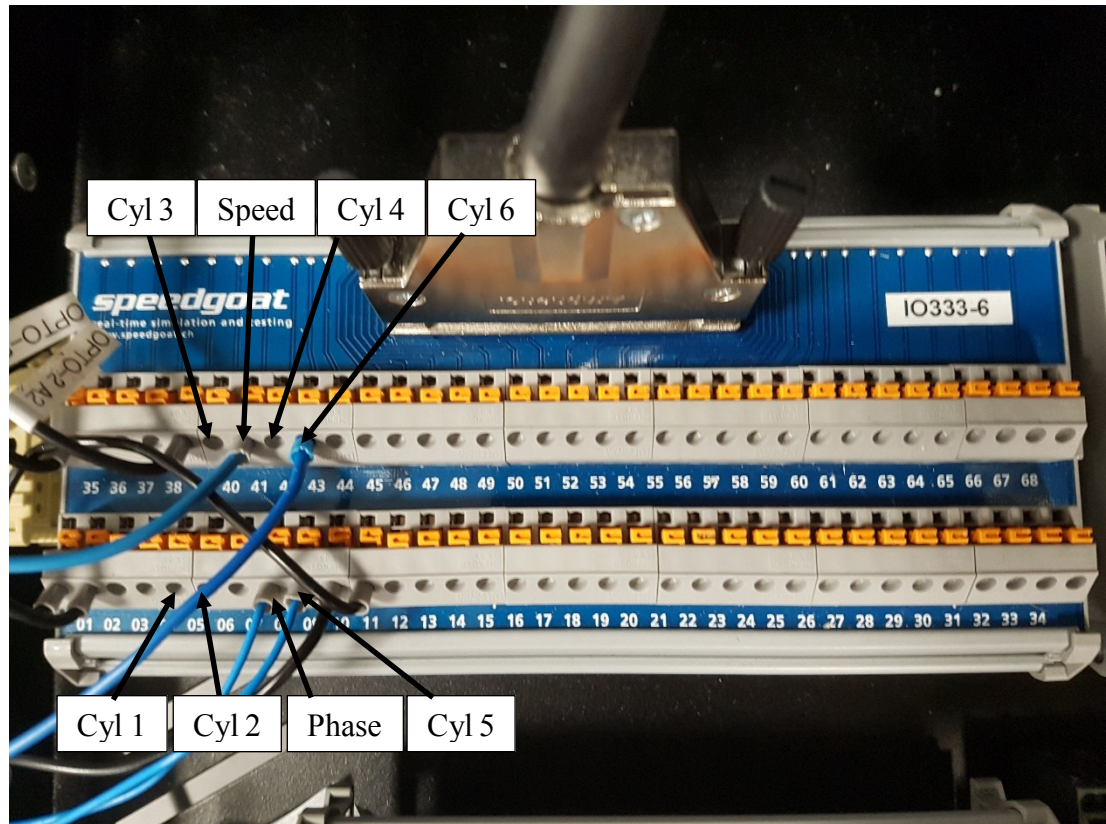


Figure 3.12 Picture of the Speedgoat I/O-card setup

The oscilloscope that was used has four channels which made measuring all the cylinders, speed and phase at the same time impossible. Instead the decision was made that channel three was going to be used for the speed and channel four for the phase and they stayed the same all the time. Channel one and two would then be used for the six different cylinders in the way that channel one would handle cylinder one, three and five while channel two would handle cylinder two, four and six. The setup of the oscilloscope can be seen in Figure 3.13

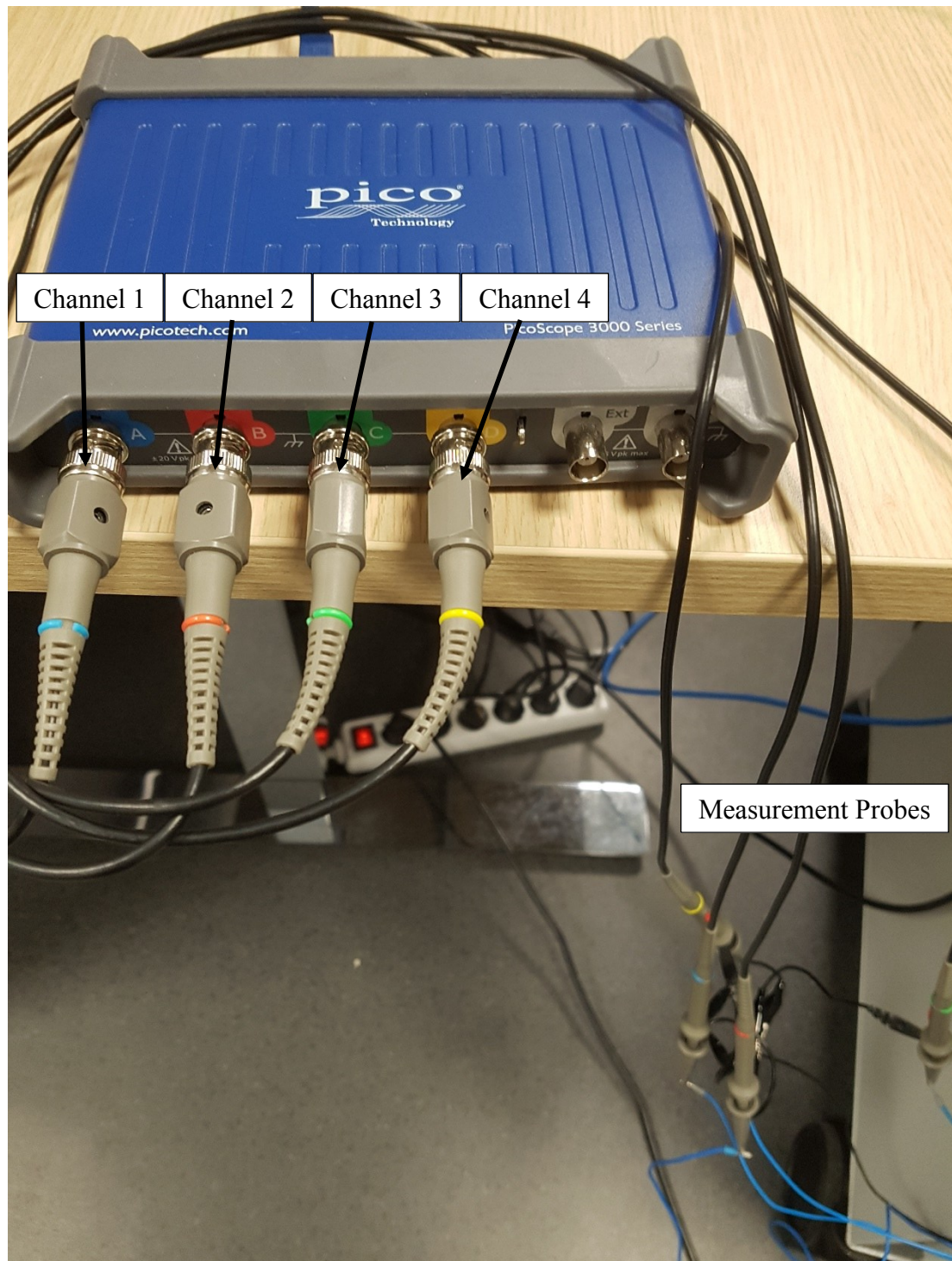


Figure 3.13 Picture of the oscilloscope setup

4 RESULTS

4.1 Accuracy of the offset of the cylinders

The cylinder injection activation output signals have been plotted alongside the engine speed and phase as a function of time. In order to validate the individual offset of the cylinders, the time between the start of the engine cycle and the start of the cylinder cycle has been measured. The simulation was operated at an engine speed of 700 rpm which equals $4.19999^\circ\text{CA}/\text{ms}$, which is needed to be able to calculate the offset in degrees. In the figures for the offset the green line represents the speed of the engine, the yellow line represents the phase of the engine and the other six colors represent the cylinder injection activation pulses.

The measurement for cylinder one can be seen in Figure 4.1. The offset between the engine cycle start and the start of the cycle for cylinder one was measured to be 0ms which then becomes an offset of 0°CA .

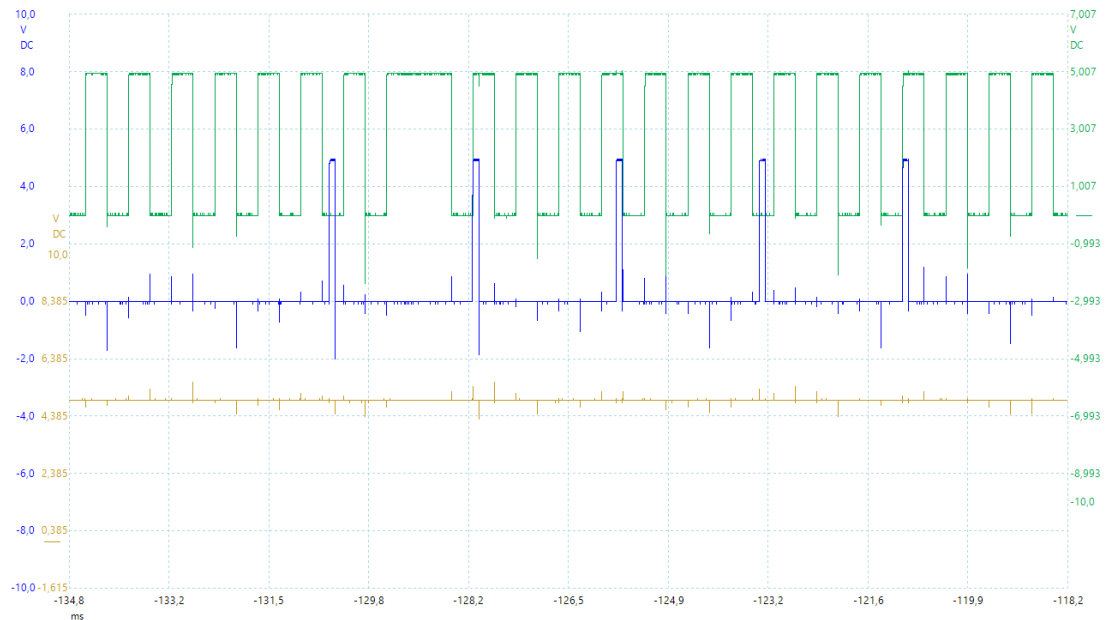


Figure 4.1 the offset of cylinder one in comparison to the start of the engine cycle

The measured offset between the engine cycle start and cylinder two cycle start was 57.1ms which equals to an offset of 239.82°CA . Since the measurement happens before the start of the cycle the actual offset of cylinder two becomes 480.18°CA .

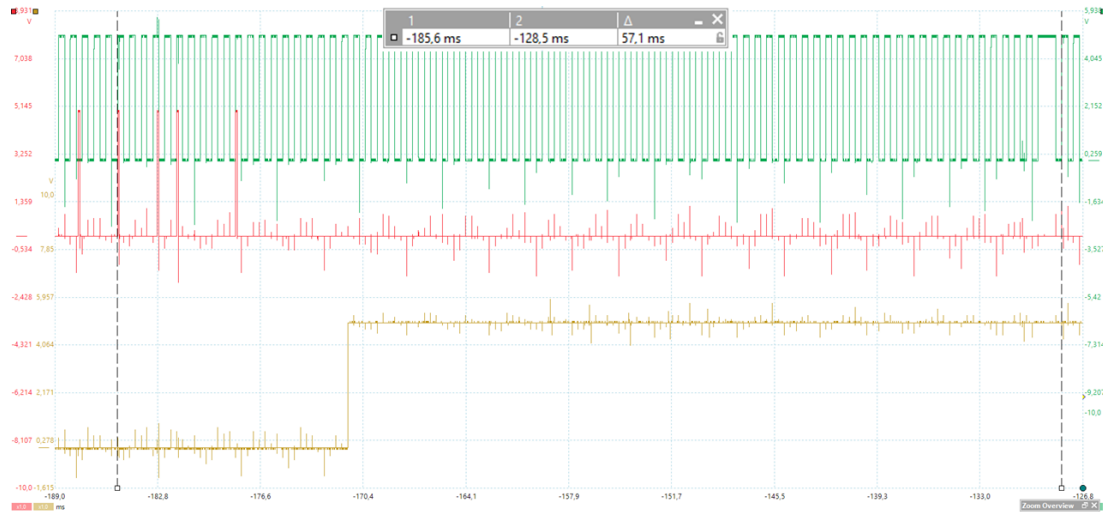


Figure 4.2 the offset of cylinder two in comparison to the start of the engine cycle

The offset measured for the start of the cycle for cylinder three was 57.1ms after the start of the engine cycle which equals to an offset of 239.82°CA .

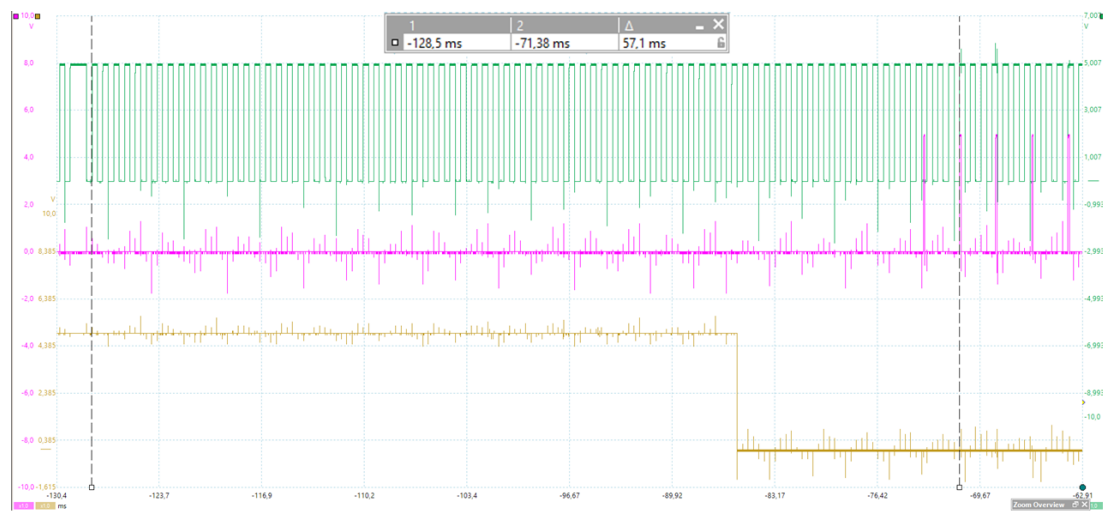


Figure 4.3 the offset of cylinder three in comparison to the start of the engine cycle

The offset for the start of the cycle for cylinder four was measured to be 28.55ms which equals to an offset of 119.91°CA, since the measurement happens before the cycle start the actual offset becomes 600.09°CA.



Figure 4.4 the offset of cylinder four in comparison to the start of the engine cycle

The offset for the start of the cycle for cylinder five was measured to be 28.55ms from the start of the cycle which equals to an offset of 119.91°CA.

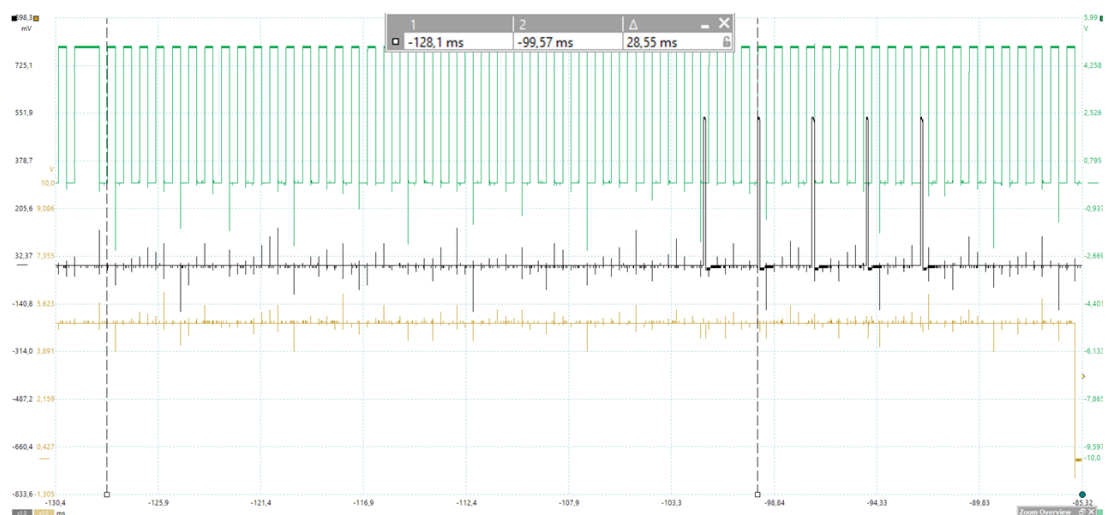


Figure 4.5 the offset of cylinder five in comparison to the start of the engine cycle

The offset for the start of the cycle for cylinder six was measured to be 0ms which equals to an offset of 0°CA, but since the offset happens during the top dead center (TDC) when the phase is low the actual offset to the start of the cycle becomes 360°CA.

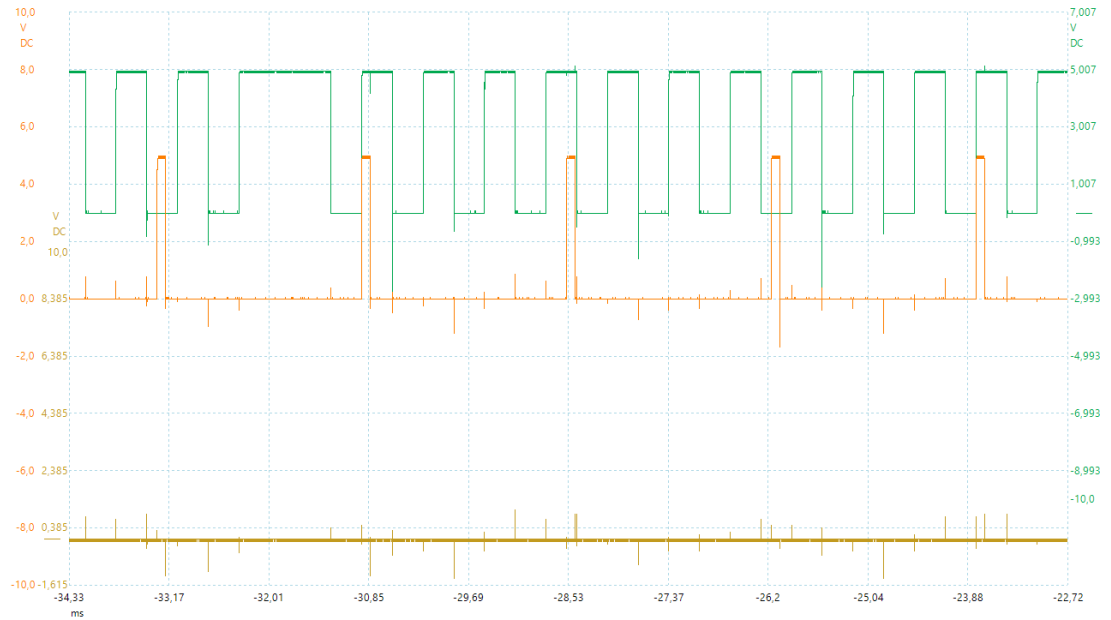


Figure 4.6 the offset of cylinder six in comparison to the start of the engine cycle

A plot of the error between the setpoint and the actual value of the start of the injection pulses of the cylinders can be seen in Figure 4.7. The pulses were measured with an interval of 20°CA.

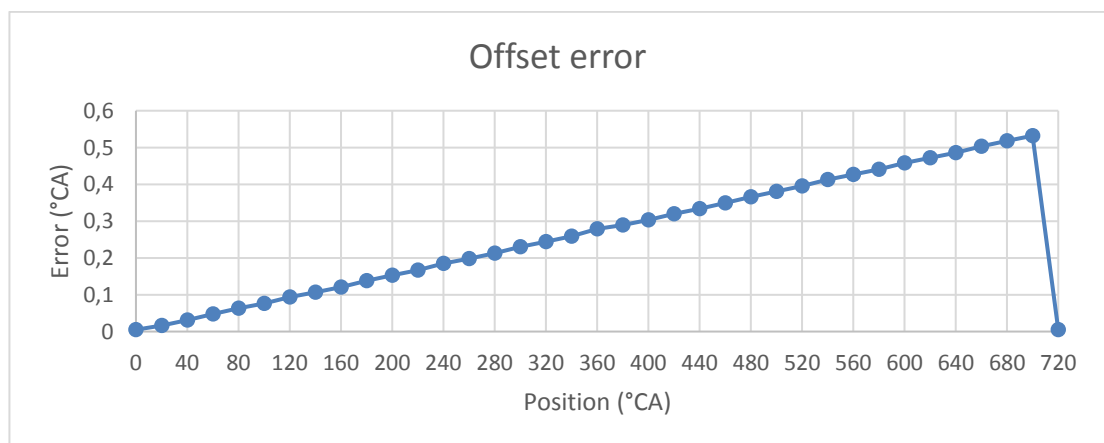


Figure 4.7 the difference in °CA between the measured offset and the desired value in a cycle

4.2 Accuracy of the demand length of the cylinders

For the measurement of the accuracy of the demand of the injection activation pulses the demand was set to last 100 μ s and was measured from the second pulse which starts at the beginning of the cycle at 0°, these configurations applied to all six of the cylinders.

The results from the simulation of the system with regards to the length of the demand cycle of the cylinders can be seen in Figures 4.8, 4.9, 4.10, 4.11, 4.12 and 4.13.

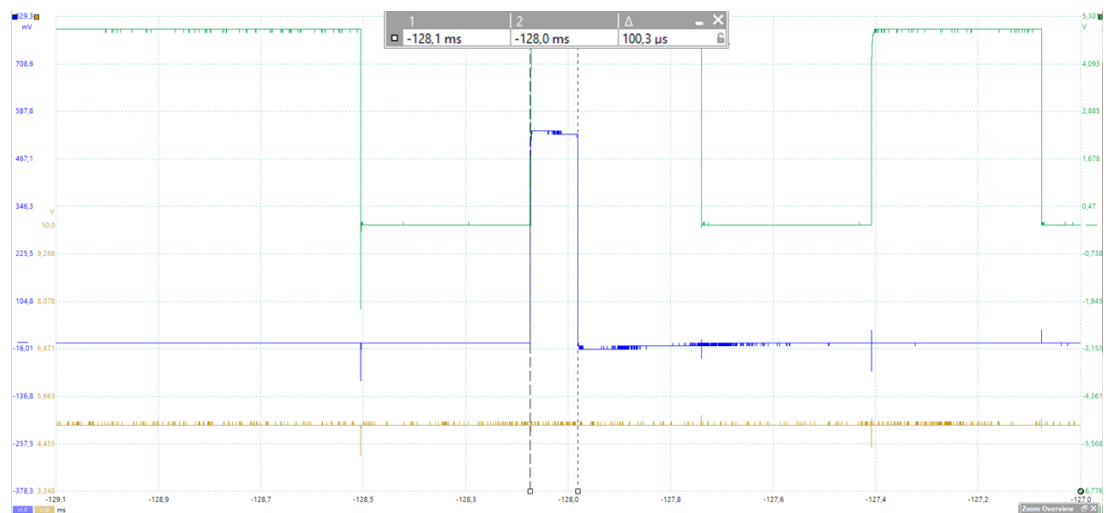


Figure 4.8 Length of the demand cycle of cylinder one

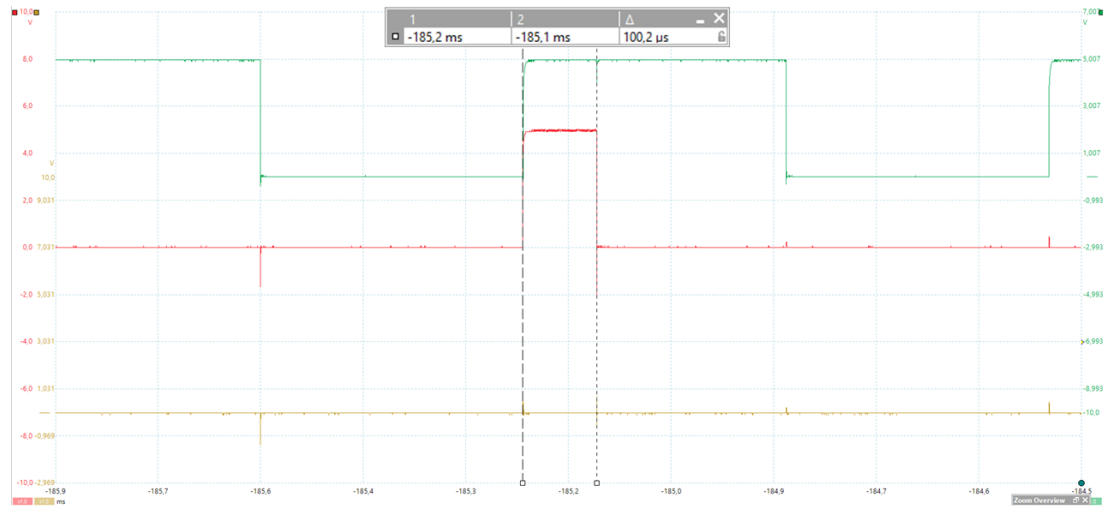


Figure 4.9 Length of the demand cycle of cylinder two

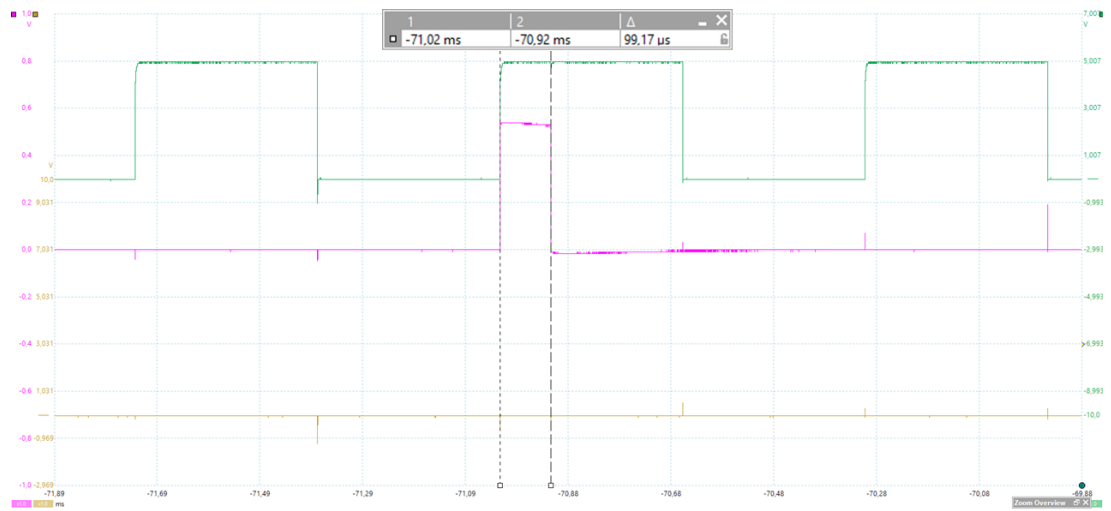


Figure 4.10 Length of the demand cycle of cylinder three

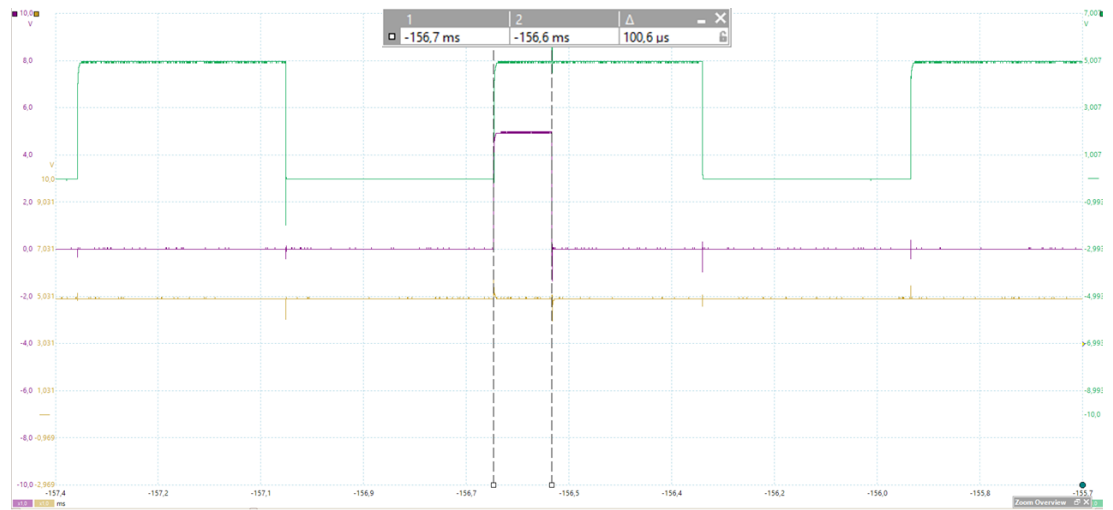


Figure 4.11 Length of the demand cycle of cylinder four

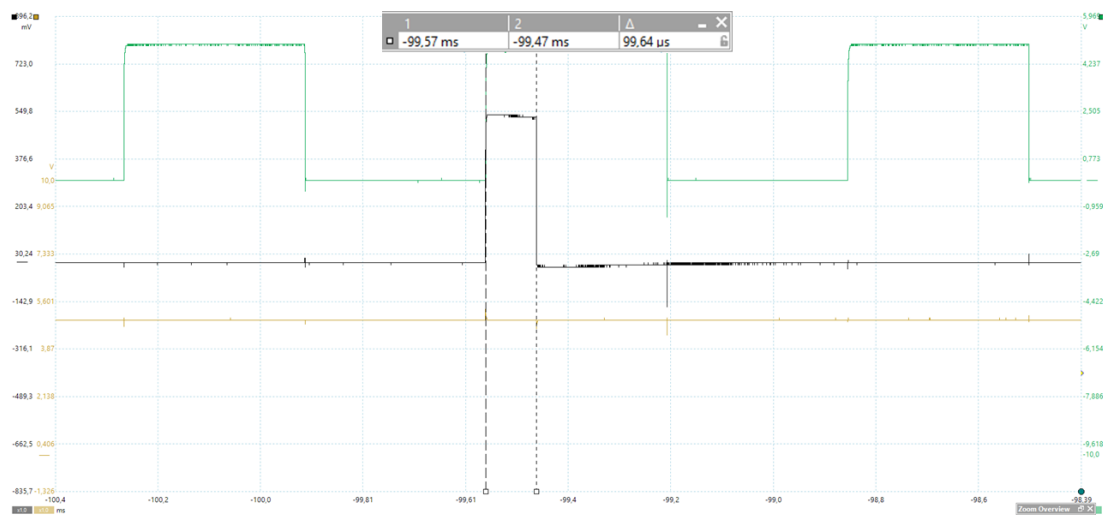


Figure 4.12 Length of the demand cycle of cylinder five

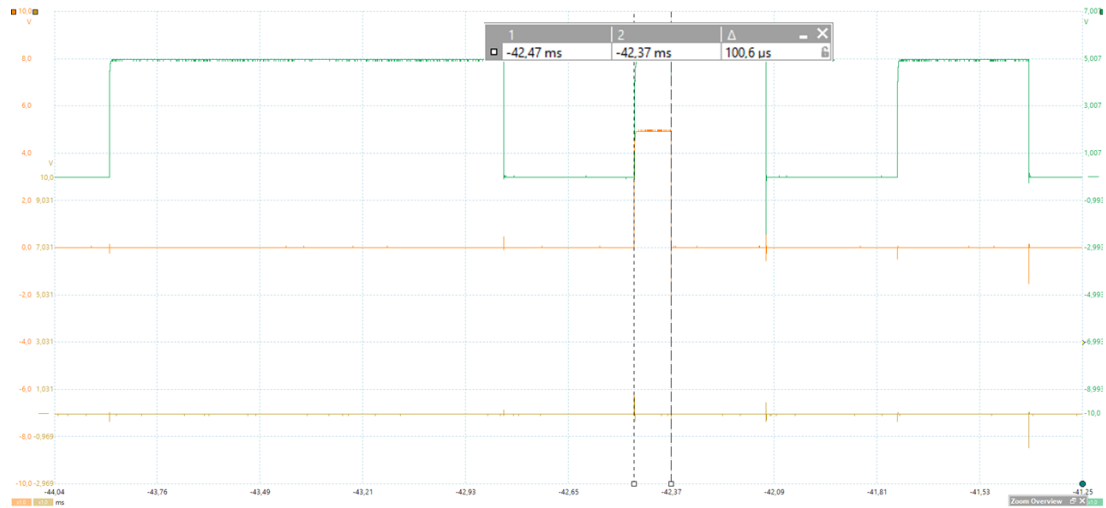


Figure 4.13 Length of the demand cycle of cylinder six

In Figure 4.14 a plot of the error between the setpoint and the actual value of the length of the demand cycle can be seen. The points that were measured are the same that was used in the test which is shown in Figure 4.7 earlier on in the document.

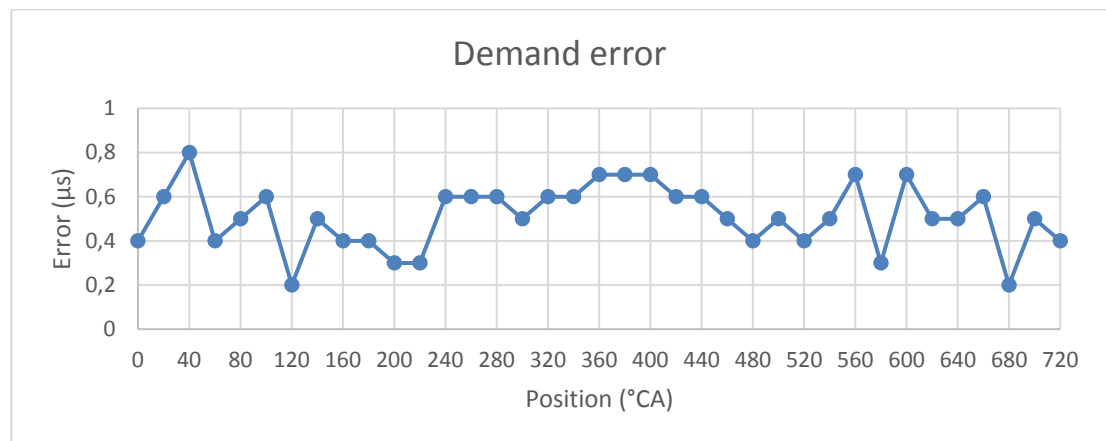


Figure 4.14 the error in percent between the measured demand and the desired value in a cycle

4.3 Third party injector driver study

Results from the study to find a suitable third party injector for the project are that there were one driver which matched the requirements set up for the project. The driver found was an IPoD coil which had been modified slightly from the original design. The company that manufactures the IPoD coil is the French company EFS.

5 DISCUSSION AND ASSESSMENT OF THE RESULTS

5.1 Simulation of the engine control system

The results from the measurement of the difference in offset were observed to be consistent down to the third decimal for all of the six cylinders that were simulated, although the results started to shift somewhat between the different cylinders when taking into account decimals beyond the first three. It is difficult to determine if it is due to the program itself or a human error during the measurement, which had to be performed manually. The measurement was realized that way because the oscilloscope software lacks a method for the program itself to measure the distance between two points in the program. The measurement was therefore executed in such a way that the signals were magnified as far as the software allowed and then the markers were placed at their respective places.

The risk for errors is always present when conducting measurements of this character, but the accuracy that was achieved in this test can be seen as reasonably reliable since it becomes difficult to determine if there is a way to eliminate the errors or if it simply is not possible when considering values as small as the ones in this experiment.

One interesting element that can be observed about the offset in the simulation which can be seen in Figure 4.7 is that it seems that the closer to the start of the engine cycle the injection point is, the smaller the error is. It then grows almost linear as the injection point moves away from the start of the cycle. Since the error increases with crank angle position it should be possible to compensate for the error by implementing a delay on the injection timing based on at which crank angle position the injection should occur. The threshold for when an error does not interfere with normal operation of the engine is 0.5°CA . With that threshold in mind it can be concluded that the operation of the engine would be disrupted by the error starting from 660°CA until the cycle resets at 720°CA . Any reasonable explanation for the occurrence of this phenomenon is yet to be found as this study concludes.

The error in the measurement of the length of the demand cycle which was presented in Figure 4.14, is much less stable than that in the offset measurement, with the error shifting between $0.2\mu\text{s}$ and $0.8\mu\text{s}$ and no logical pattern to be identified, unlike the case with the offset. When recalculating this error interval into degrees crank angle the largest one being $0.8\mu\text{s}$ becomes 0.00336°CA , which when considering the 0.5°CA threshold will have no effect on the normal performance of the engine.

5.2 Third party injector driver study

The IPoD coil did not originally have the required power to be able to be used for this study, but EFS investigated the issue and drafted an offer for an IPoD coil that would be able to deliver the required power. Unfortunately, it was not possible in the scope of this study to acquire an IPoD coil and this then in turn led to the testing of the control system to be rather more limited than what was originally planned.

6 CONCLUSIONS AND RECOMMENDATIONS

In this thesis a literature study and investigation on the web were rendered to try to find an injector driver that could be used in the study. However, in the end it could be concluded that the only injector driver which fulfilled the hardware requirements was not feasible to be used in the study due to an extremely high cost and long delivery time.

An investigation was also conducted into the possibility of using Wärtsilä's existing CCM-30 driver in the study. It was determined that it would be possible to modify the CCM-30 in such a way that it would be possible to use it. Due to the complexity and size of the modification needed, it was not feasible to include that part in this study, and the work was therefore outsourced within the company. This in turn led to a situation where the timetables for this study and the modification of the CCM-30 did not match, and it became clear over time that it would not be possible to test the full system with the CCM-30 in this study.

A control system for the injector driver was developed in Matlab Simulink in which the inputs are the engine speed, the position and the demand and timing of the injection points. The output from the system is a fuel injection activation signal in the form of a high-low signal (0-5V) for each of the six cylinders of the engine which the system is developed for. The parameters of the control system which the user has control over are the demand and the timing inputs.

One very interesting outcome of the simulation of the control system was the fact that the error in the timing of the injection pulses not only was consistent across all six cylinders but also was almost completely linear across the complete cycle. However, these kinds of characteristics almost make one wonder if there is something wrong with the model itself considering the "neatness" of the errors. This is something that could be investigated further when using the model in the future.

A more expected result was the fact that the error in the measurement of the length of the injection pulse against the setpoint of the demand was very random with no recognizable pattern whatsoever.

The next step in the testing of this model would have been to connect it to a CCM-30 and have the model control the injection drivers in the CCM-30 module. Due to outside circumstances it was not possible to realize those tests within the scope of this study. This would then be the next logical step in the future research of the functionality of this model.

Considering the state that the control system is in now, it is too early to determine if it is feasible for Wärtsilä to start using this system on a larger scale. But as a means for testing different concepts on a relatively small scale in an office or laboratory setting, the control system could definitely be used, when compatibility with the modified CCM-30 has been determined in future tests and simulations. The error in the injection timing should also be reduced below 0.5°CA for every crank angle position in the cycle, so that it will not interfere with the normal operation of the engine.

SWEDISH SUMMARY

Implementation av ett snabbt styrsystem för att styra aktuatorer i en medelvarvig förbränningsmotor

Inledning

Nu för tiden försöker man få bukt med och minska på alla skadliga utsläpp från speciellt den marina industrin genom alltmer strikta förordningar som begränsar kraftigt den tillåtna mängden av dessa utsläpp från fartygsmotorer. Några av utsläppen som man speciellt fokuserar på är kväveoxider, svaveloxider, kolmonoxid samt oförbrända kolväten.

För att minska på utsläppen kan man använda olika metoder, t.ex. förbättra förbränningsprocessen i de motorer som används i fartygen och på så sätt minska på utsläppen. För att förbättra förbränningen kan man i stort sett använda en metod för varje enskild del i förbränningsprocessen. I denna studie är fokus på bränsleinsprutningen i cylindern och ett försök att förbättra snabbheten på kommunikation mellan de elektriska signaler som går från styrsystemet till drivsteget i injektorn har gjorts. På så sätt kan man minska på fördröjningen i kommunikationen mellan styrsystem och drivsteget, en fördröjning som kan ställa till med stora problem ifall den blir för lång.

Behovet för studien uppstod när Advanced concepts-arbetslaget på Wärtsilä var i behov av ett flexibelt och lätthanterligt styrsystem för att kontrollera bränsleinjektorer i en testmiljö. Styrsystemet ska vara kapabelt att hantera väldigt snabba injektorer, vilket ställer till med en del problem i nuläget eftersom de injektorer som används nödvändigtvis inte är gjorda för det ändamålet. Detta har lett till att det inte har funnits någon färdig lösning att tillgå på detta problem.

Syfte

Det övergripande syftet med detta examensarbete är att undersöka vilka möjligheter det finns att utveckla ett snabbt och lätthanterligt styrsystem för att kontrollera drivsteget i bränsleinsprutningsprocessen. Styrsystemet kommer att utvecklas i Matlab Simulink, som är ett verktyg med vilket man bygger en grafisk simulationsmodell med hjälp av olika block som innehåller en fördefinierad Matlab-kod.

En litteraturstudie kommer också att göras för att undersöka huruvida det finns något drivsteg från en utomstående part som kan användas i studien. Om ett sådant drivsteg hittas kommer styrsystemet att anpassas efter det. Om inget sådant drivsteg hittas kommer det att undersökas ifall det är möjligt att använda Wärtsiläs eget drivsteg CCM-30 för studien. När styrsystemet är färdigställt kommer hela kedjan att testas med hjälp av en realtidshårdvara från Speedgoat.

Metod

Mätningen av motorns position sker så att cykeln startar vid 0 grader och sedan räknar modellen uppåt till 360 grader där nästa top dead center (TDC) sker. Sedan börjar den räkna ner från -360 grader ända tills den når 0 grader igen och cykeln är slutförd. Modellen består av två 360 graders cykler för att den är gjord för en fyrtaktsmotor och har således två fulla rotationer i varje cykel.

Modellen genererar också en återställningspuls för varje cylinder som används för att signalera att cylinderns cykel startar. Detta görs genom att signalen för motorns position jämförs i simulink när den är större än 0 och samtidigt mindre än 0. Dessa två signaler leds sedan in i en logisk AND-funktion som kommer att ge ut en logisk 1 när cylinderns cykel påbörjas. Denna signal blir då modellens återställningspuls för cylindrarna.

Simulink-modellen har två parametrar som är definierade av användaren; längden på signalen som aktiverar bränsleinsprutningen samt tajmningen för samma signal.

Tajmningen innebär vid vilken tidpunkt signalen ska aktiveras och längden hur länge signalen ska vara aktiverad. Tajmningen anges i grader vevvinkel och längden anges i mikrosekunder. Dessa två parametrar är sparade i Simulink-modellen i form av vektorer där alla vektorer har fem element var, för att möjliggöra fem insprutningar per cykel. Modellen har sammanlagt tolv vektorer, en vektor för tajmningen och en för längden på signalen, för motorns samtliga sex cylindrar.

Vektorerna leds därefter in i nästa del av modellen där de sedan separat leds in i sina respektive cylindrars delsystem. Inne i delsystemen delas längd- och tajmningvektorer upp så att varje enskilt element blir en egen signal. Dessa tio signaler leds därefter tillsammans med signaler för positionen och positionssynkronisering in i nästa steg i modellen, vilket är ett Stateflow-program som har till uppgift att aktivera utgångarna från modellen vid rätt tillfälle.

Stateflow-programmet börjar med att verifiera att modellen och motorn är synkroniserade. Detta görs genom att modellen väntar på att positionssynkroniseringen ska få värdet 1 vilket signalerar att de är synkroniserade. Därefter väntar programmet även på en återställningssignal som berättar för programmet att cylindern har påbörjat en ny cykel. Sedan går programmet vidare och väntar på att cykeln ska nå den första tajmningspositionen som är definierad i tajmningsvektoren. När den har nått den går den vidare till nästa steg där den kollar om signallängdsvektorn har något värde i sitt första element. Om längdsignalen har ett värde blir utgången från Stateflow-programmet som är aktiveringssignalen för bränsleinsprutningen 1 och programmet börjar räkna upp från 1 ända tills det har nått samma värde som har sparats i vektorn. Ifall värdet på längdsignalen är 0 hoppar programmet över detta steg och går direkt vidare och väntar på nästan tajmningsposition. Detta repeteras därefter för alla fem bränsleinsprutningspunkter och sedan återgår programmet till steget där det väntar på återställningssignalen för cylindern. Det enda undantaget är om positionssynkroniseringen under något av dessa steg skulle bli 0, då går programmet ända tillbaka till början av programmet och väntar tills den signalen blir 1 igen.

En syntes av Simulink-modellen behöver göras för att det ska vara möjligt att

importera den till FPGA-kortet som sitter i hårdvaran från Speedgoat. Denna syntes görs med hjälp av ett verktyg som heter HDL advisor som finns integrerat i Simulink.

Speedgoat-hårdvaran är monterad inne i ett skåp för att underlätta testning i kontorsutrymmen. Datorn som kör programmet med vilket hårdvaran styrs kopplas ihop med hårdvaran via en ethernetkabel. Realtidshårdvaran är i sin tur kopplad till ett externt I/O-kort som används för att kunna mäta in- och utsignaler från hårdvaran.

För att mäta utsignalerna från kortet, det vill säga bränsleinsprutningssignalerna, hastigheten och vilken fas motorn är i, används ett oscilloskop. Oscilloskopet hade bara fyra mätkanaler så hastigheten och fasen mättes under samtliga tester medan de andra utsignalerna, det vill säga cylindrarna, delades upp och mättes två och två, vilket resulterade i totalt tre olika mätningar.

Resultat och diskussion

Resultatet av försöket att hitta ett drivsteg från en utomstående part blev att det fanns ett drivsteg som kunde användas i studien, en IPoD coil som tillverkas av det franska företaget EFS. Drivsteget kunde dock inte i originalutförande användas för studien men EFS undersökte saken och kom fram till att de kunde erbjuda en omändrad IPoD coil som var tre gånger så stark som en originalversion. Denna version skulle då vara tillräckligt stark för att användas i detta projekt.

Simuleringen av styrsystemet kördes med en motorhastighet på 700 varv per minut, vilket blir 4,2 grader vevvinkel per millisekund som i sin tur betyder att en hel cykel på 720 grader vevvinkel blir cirka 171,4 millisekunder lång. Detta samband är nödvändigt när man ska räkna om fördröjningen och eventuella fel till grader från millisekunder, som är enheten man får ut från mätprogrammet.

Resultatet av styrsystemets testning visades i två grafer. Den ena var en graf över hur stor skillnaden var mellan börvärdet för tajmningen av bränsleinsprutningssignalen och det verkliga uppmätta resultatet. Mätpunkterna togs med 20 graders mellanrum med start från 0 grader och sedan upp till 720 grader för att få med hela cykeln. Den

andra var en snarlik graf där det för samma punkter som i föregående mätning visades skillnaden i börvärde och det verkliga värdet för längden på bränsleinsprutningssignalen.

I slutändan blev det bestämt att IPoD coilen inte skulle användas, främst för att anskaffningskostnaderna för det totala antalet som skulle ha behövts hade blivit alltför stort. Leveranstiden för drivstegen skulle också blivit för lång.

Resultatet från mätningen av tajmningen gav ett aningen oväntat fel i det verkliga värdet i förhållande till börvärdet. Felet var näst intill obetydligt när det rörde sig om grader, vilka var väldigt nära motorns TDC, men när man förflyttade sig längre bort från TDC så växte felet nästan helt linjärt ända tills cykeln var slutförd, för att sedan upprepas igen. Någon tillfredställande förklaring till detta fenomen har hittills inte hittats och det skulle säkerligen vara värt att undersöka vidare i framtiden.

I mätningen av längden på bränsleinsprutningspulsen återfanns inget lika linjärt fel, utan där var felet lite mer som förväntat och var väldigt slumpmässigt till sin natur. Inga tydliga tecken på vad som kunde tänkas påverka felet kunde urskiljas.

Framtida forskning skulle vara att ta Simulink-modellen som utvecklats här och koppla ihop den med en CCM-30 och testa ifall det är möjligt att styra den med hjälp av detta styrsystem. Ytterligare vore det värt att undersöka felet som uppdagades när det gäller att testa tillförlitligheten i tajmningen för insprutningssignalerna, för att ta reda på ifall felet härstammar från Simulink-modellen eller om det är mätutrustningen från Picoscope som är ursprunget.

7 REFERENCES

- Bishop R. & Dorf R. (2011). *Modern Control Systems*. New Jersey: Pearson Higher Education.
- Brauer J. R. (2006). *Magnetic Actuators and Sensors*. Hoboken, New Jersey: John Wiley & Sons Inc.
- Breitzman R. C. (1985). Development of a custom microprocessor for automotive control. *IEEE Control Systems Magazine*, 5(2), pp. 23-28.
- Brown S. & Rose J. (2002). Architecture of FPGAs and CPLDs: A tutorial. *IEEE Design and Test of Computers*, 13(2),
- Chen K. S. (2000). Simultaneous Reduction of NO_x and Particulate Emissions by Using Multiple Injections in a Small Diesel Engine. *SAE Technical Paper*, p. 12.
- Farrugia M. et al. (2015). *Development of a programmable engine control unit to multi-fuel capabilities, progression to diesel*. Monastir, 2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA).
- Golnaraghi F. & Kuo B. (2009). *Automatic Control Systems*. New Jersey: John Wiley & Sons, Inc.
- Grepl R. 2011. *Real-Time Control Prototyping in MATLAB/Simulink: Review of tools for research and education in mechatronics*. Istanbul, Turkey
- Heywood J. B. (1988). *Internal Combustion Engine Fundamentals*. Singapore: McGraw-Hill.
- Hilbert J. (1991). *Application Specific Integrated Circuit (ASIC) Technology*. San Diego: Academic Press, Inc.
- Hölttä V. Palmroth L. & Eriksson L. (2004). Rapid control prototyping tutorial with application examples. January.
- Khajepour A. Fallah S. & Goodarzi A. (2014). *Electric and Hybrid Vehicles - Technologies, Modeling and Control: A Mechatronic Approach*. :John Wiley & Sons.
- Khan J. (2016). *Rapid Control Prototyping (RCP) solutions for the validation of motor control applications*. Kollam, India
- Kiencke U. & Nielsen L. (2010). *Automotive Control Systems*. Berlin: Springer.
- Krishna, C. M. (1999). Real-Time Systems. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 27 December, pp. 262-270.

Laplante P. A. & Ovaska S. J. (2012). *Real-Time Systems Design and Analysis: Tools for the Practitioner*. New Jersey: John Wiley & Sons, Inc.

Lucchini T. et al. (2010). Experimental and Numerical Investigation of High-Pressure Diesel Sprays with Multiple Injections at Engine Conditions. *SAE Technical Paper*, p. 14.

Mendonca L. S. Luceiro D. D. Martins M. E. S. & Bisogno F. E. (2017). *Development of an engine control unit: Implementation of the architecture of tasks*. Toronto, 2017 IEEE International Conference on Industrial Technology (ICIT).

Nise N. S. (2011). *Control Systems Engineering*. Jefferson City: John Wiley & Sons, Inc.

Okude K. et al. (2007). Effects of Multiple Injections on Diesel Emission and Combustion Characteristics. *SAE Technical Paper*, p. 12.

Posey N. Wu T. Wolff M. & Zumberge J. (2012). *Modeling and control scheme design of a solenoid-actuated fuel injection system*. Dayton, OH, USA

Satkoski C. Ruikar N. Biggs S. & Shaver G. (2011). *Cycle-to-cycle Estimation and Control of Multiple Pulse Profiles for a Piezoelectric Fuel Injector*. San Francisco, CA, USA

Trimberger S. (1994). *Field-Programmable Gate Array Technology*. New York: Kluwer Academic Publishers.

Underwood K. (2004). *FPGAs vs. CPUs: Trends in Peak Floating-Point Performance*. Monterey, California, Association for Computing Machinery, pp. 171-180.

Young H. D. Freedman R. A. & Lewis Ford A. (2011). *University Physics with modern physics*. 13th ed. San Francisco: Pearson Education Inc.

Young S. L. Bongeon J. & Soohee H. (2017). A Light-Weight Rapid Control Prototyping System Based on Open Source Hardware. *IEEE Access*, 5(8), pp. 11118-11130.

Zheng G. Wang F. Wang S. & Gao W. (2014). Urea SCR System Development for Large Diesel Engines. *SAE Technical Paper*, p. 9.